AD A083313

ENGINEERING

LEVEL II

USCIPI Report 950

UNIVERSITY OF SOUTHERN CALIFORNIA

DESIGN OF SVD/SGK CONVOLUTION FILTERS FOR IMAGE PROCESSING

by

Sang Uk Lee

January 1980

DTIC
ELECTE
APR 23 1980
S D
C

Department of Electrical Engineering
Image Processing Institute
University of Southern California
Los Angeles, California 90007

Sponsored by
Defense Advanced Research Projects Agency
Contract No. F-33615-76-C-1203
DARPA Order No. 3119

DDC FILE COPY

This document has been approved for public release and sale; its distribution is unlimited.

IPI

IMAGE PROCESSING INSTITUTE

80 4 21 079

USCIPI Report 950

DESIGN OF SVD/SGK CONVOLUTION FILTERS FOR IMAGE PROCESSING

by

Sang Uk Lee

January 1980

Department of Electrical Engineering
Image Processing Institute
University of Southern California
Los Angeles, California 90007

Research sponsored by:
Defense Advanced Research Projects Agency
Contract No. F-33615-76-C-1203
DARPA Order No. 3119

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. ADA083 313	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESIGN OF SVD/SGK CONVOLUTION FILTERS FOR IMAGE PROCESSING		5. TYPE OF REPORT & PERIOD COVERED Technical Report, January 1980
		6. PERFORMING ORG. REPORT NUMBER USCIPI [illegible] 950
7. AUTHOR(s) SANG UK LEE		8. CONTRACT OR GRANT NUMBER(s) F33615-76-C-1203, DARPA Order 3119
9. PERFORMING ORGANIZATION NAME AND ADDRESS Image Processing Institute University of Southern California Los Angeles, California, 90007		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS DARPA Order No. 3119
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Blvd. Arlington, Virginia 22209		12. REPORT DATE January 1980
		13. NUMBER OF PAGES 198
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Air Force Avionics Laboratory U.S. Air Force Air Force Systems Command Wright-Patterson AFB, OHIO 45433		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for release: distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Image Processing, Digital Filtering, Image Restoration, Singular-value Decomposition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This dissertation describes a special-purpose signal processor for performing two-dimensional convolution with a minimum amount of hardware using the concepts of singular value decomposition (SVD) and small generating kernel (SGK) convolution. The SVD of an impulse response of a two-dimensional finite impulse response of a two-dimensional finite impulse response (FIR) filter is employed to decompose a filter into a sum of two-dimensional separable linear operators. These linear operators are themselves decomposed into a sequence → next page (continued)

DD FORM 1 JAN 73 1473

291 141

of small kernel convolution operators. The SVD expansion can be truncated to a relatively few terms without significantly affecting the filter output.

A statistical analysis of finite word-length effects in SVD/SGK convolution is presented. Two important issues, related to the implementation of the filters in cascade form, scaling and section ordering, are also considered.

Computer simulation of image convolution indicates that 12 bits are required for the SGK/SVD accumulator memory and 16 bits are required for quantization of filter coefficients to obtain results visually indistinguishable from full precision computation. A normalized mean square error between the SVD/SGK processed output and the direct processed output is chosen as an objective criterion function. It is shown that a subjective visual improvement is obtained by resetting the output mean to be equal to the input mean.

The transformation technique developed for the one-dimensional case is used to parametrically modify the cutoff frequency of a baseline SVD/SGK convolution filter. A detailed discussion of the one-dimensional case is presented, and its applicability to SVD/SGK convolution filters is described.

Accession For
NTIS GRA&I
DDC TAB
Unannounced
Justification
By
Distribution/
Availability Codes
Dist | Avail and/or special
A

ACKNOWLEDGEMENTS

I would like to express my sincere appreciation to Dr. William K. Pratt, chairman of my dissertation committee, for suggesting this topic and for his guidance and constant encouragement throughout this work. This work would not have been completed without his support.

I am deeply grateful to Dr. J.F. Abramatic, who was a research scientist at USC, for the valuable criticism, suggestions and countless discussions in the course of this work.

Thanks are also extended to the rest of the committee, Professor Alexander A. Sawchuk and Professor J. Milstein, for their assistance and contributions.

I am also indebted to the staff of the Image Processing Institute for their support given in preparation of this work, especially Ms. Hilda Marti for her excellent typing.

Finally, my special thanks go to my wife, Joo Won, for her years of sacrifice, patience and understanding.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

This dissertation describes a special-purpose signal processor for performing two-dimensional convolution with a minimum amount of hardware using the concepts of singular value decomposition (SVD) and small generating kernel (SGK) convolution. The SVD of an impulse response of a two-dimensional finite impulse response (FIR) filter is employed to decompose a filter into a sum of two-dimensional separable linear operators. These linear operators are themselves decomposed into a sequence of small kernel convolution operators. The SVD expansion can be truncated to a relatively few terms without significantly affecting the filter output.

A statistical analysis of finite word-length effects in SVD/SGK convolution is presented. Two important issues, related to the implementation of the filters in cascade form, scaling and section ordering, are also considered.

Computer simulation of image convolution indicates that 12 bits are required for the SGK/SVD accumulator memory and 16 bits are required for quantization of filter coefficients to obtain results visually indistinguishable from full precision computation. A normalized mean square

error between the SVD/SGK processed output and the direct processed output is chosen as an objective criterion function. It is shown that a subjective visual improvement is obtained by resetting the output mean to be equal to the input mean.

The transformation technique developed for the one-dimensional case is used to parametrically modify the cutoff frequency of a baseline SVD/SGK convolution filter. A detailed discussion of the one-dimensional case is presented, and its applicability to SVD/SGK convolution filters is described.

CHAPTER 1

INTRODUCTION

During the last decade, the field of digital signal processing has been extremely dynamic and active. There have been many applications of digital signal processing techniques in digital communication, seismic processing, radar processing, sonar processing, speech processing, and image processing.

One of the important areas in digital signal processing is digital filtering. The term "digital filtering" can be viewed as a computational process or algorithm by which a sampled signal or a sequence of numbers, acting as an input signal, is transformed into a second sequence of numbers called the output. There are two major types of digital filters: infinite impulse response (IIR) filters and finite impulse response (FIR) filters. Digital filtering is mainly concerned with filter design and its implementation.

If the present output of a system is calculated from the past, present, and, in the noncausal case, future inputs, the system is called nonrecursive. If the present

output of a system is calculated from the past and present inputs and outputs, the system is called recursive. In both recursive and nonrecursive systems, the relation between an input sequence x(n) and an output sequence y(n) can be characterized by a difference equation of the form

$$y(n) = \sum_{k=0}^{M} a_k f(n-k) - \sum_{k=0}^{N} b_k y(n-k) \qquad (1\text{-}1)$$

Conceptually, M and N can be finite or infinite. A system in which $b_k = 0$, for $k = 1,\ldots,N$, is nonrecursive, and can be implemented by an FIR filter. The system in which $N \geq 1$ and b_k is not zero is recursive, and it can be implemented by an IIR filter. Choosing between an FIR filter and an IIR filter depends upon the relative advantages and disadvantages the filter offers for a specific problem.

Signal processing is, of course, not limited to one-dimension. Many signals are inherently two-dimensional; thus, two-dimensional signal processing techniques are required. Image data is a typical two-dimensional signal. Digital filtering with FIR filters has many applications in image processing. For instance, image restoration to remove blur and to suppress noise generally requires digital filtering. In most cases, digital filtering requires implementation of a two-dimensional convolution.

The term "implementation" means that the algorithm is either written in a computer language for a general-purpose computer or is realized with special-purpose hardware. In general, the implementation of two-dimensional convolution in image processing has been confined primarily to computer programs with a general-purpose computer, where virtually unlimited memory, processing capability, and time, are readily available. But the required processing time is often quite enormous because of the huge amounts of data to be processed and the restricted input-output transfer rate between the computer and display. An image size of 512x512 pixels is common in image processing. An alternative to the use of a general-purpose computer is to utilize Integrated Circuit (IC) technology. Recent advances in IC technology now make the realization of a real-time signal processor capable of performing two-dimensional convolution practical. High speed digital multipliers, memory and display circuitry are now commercially available. As a result, significantly more sophisticated algorithms can now be chosen for problem solving. The trend is to develop special-purpose signal processors to take advantage of recent developments in digital circuits [1-1 to 1-3]. In the design of such a special-purpose signal processor, speed, complexity, power consumption, computing capability, and cost, are all factors to be considered.

Recently, a technique called small generating kernel (SGK) convolution has been proposed [1-4]. SGK convolution is a processing technique for performing convolution on a two-dimensional data array by sequentially convolving the array with a small size convolution kernel, say 3x3. This idea was first suggested by Mersereau et al. [1-5] and generalized later by Abramatic and Faugeras [1-4]. Since a large size kernel convolution is performed by a sequential small size kernel convolution, and the implementation is highly modular, the SGK approach makes the hardware implementation quite appealing if a proper design procedure to specify the small size kernel operators is found.

In the one-dimensional case, any impulse response can be decomposed into small size convolution operators, typically 3x1. This property can be seen in the cascade form for FIR filters. But, theoretically, exact decomposition of a large size convolution operator into small size convolution operators is impossible in the two-dimensional case. This is the reason why the design procedure for SGK convolution leads to a complicated and time-consuming optimization problem. The inherent difficulty in finding small size convolution operators motivates the development of a new algorithm for the two-dimensional convolution. The proposed SVD/SGK convolution method also makes use of SGK convolution, however, the size of small size convolution operators is

3x1, rather than 3x3.

This dissertation describes a special-purpose signal processor with a minimum amount of hardware for performing two-dimensional convolution using the concepts of singular value decomposition (SVD) and SGK convolution. To extend the usefulness of SGK convolution, two-dimensional FIR filters of size $N_1 x N_2$ are decomposed into a sum of two-dimensional separable filters by means of the SVD of their impulse response matrix $\underline{H}$. The SVD expansion can be truncated to K terms ($K \leq R$, where R is a rank of $\underline{H}$), without significantly affecting the output of the filter. Whenever the two-dimensional FIR filter is separable, the convolution can be performed by one-dimensional processing. This is a reason why the SVD expansion can be very useful for implementing two-dimensional nonseparable filters. It was noted that each one-dimensional FIR filter can be realized as a cascade of second-order SGK filters. Thus, it is possible to implement a two-dimensional convolution by a sequential convolution with one-dimensional second-order SGK filters. As an example, one can think of using such a convolution technique for convolving images at real-time rates on an image display system.

When a digital signal processing algorithm is implemented with a special-purpose signal processor, account must be taken of the errors caused by finite

word-length in representing filter coefficients and signal values. Implementation with finite word-length can be modeled by injecting white noise into signals whenever a rounding operation is performed. The goal of this error analysis is to minimize the required word-length subject to some reasonable error tolerance. The problem is to determine the best ordering and scaling procedure in order to minimize the required word-length. To solve these two problems, we show that how the theory, for the one-dimensional case, can be modified to the two-dimensional case.

The second issue investigated in this dissertation is parametric design. The concept of parametric design is to generate a class of two-dimensional FIR filters with a variable cutoff frequency from previously designed baseline SVD/SGK convolution filters. In the case of one-dimensional FIR filters, Oppenheim et al. [1-6] have proposed a transformation for designing a variable cutoff digital filter. But, very little work has been reported in the two-dimensional case. It is shown that the cutoff frequency of a SVD/SGK convolution filter can be varied by the use of a one-dimensional transformation. It is believed that such a variable cutoff SVD/SGK convolution filter has numerous applications in image processing. Adaptive filtering will be very useful in image restoration. For example, the cutoff frequency of a Wiener

filter could be changed, and an observer could effectively examine the processed image in real-time.

This dissertation consists of seven chapters. A review of SGK and SVD/SGK convolution is presented in Chapter 2. Chapter 3 discusses the effect of using fixed-point arithmetic. This chapter includes a derivation of the noise formula to predict total roundoff noise. Scaling and section ordering for SVD/SGK convolution are described in Chapter 4. In Chapter 5, a series of experimental results based on computer simulation is presented. Among these results is the confirmation of the derived noise formula using a random number array as an input. A simple technique to reduce the normalized mean square error (NMSE) between the SVD/SGK processed output and the direct processed output is also given. The effectiveness of this technique is demonstrated visually. Chapter 6 deals with the parametric design of SVD/SGK convolution filters. A detailed discussion of the one-dimensional case is presented, and its applicability to SVD/SGK convolution filters is described. Several design examples for two-dimensional, as well as one-dimensional cases, are shown in this chapter. Finally, Chapter 7 discusses the conclusion and possible extension of this work.

CHAPTER 2

SEQUENTIAL CONVOLUTION TECHNIQUES

2.1 Introduction

Two-dimensional convolution has found numerous applications within the field of two-dimensional signal processing [2-1,2-2]. For example, image enhancement, image restoration, and digital filtering generally require two-dimensional convolution. Referring to Fig. 2-1, an output array G(j,k) is obtained by convolving the input array F(j,k) with the impulse response of the system H(j,k). The two-dimensional direct convolution algorithm can be expressed by the double sum

$$G(j,k)=F(j,k)\circledast\circledast H(j,k)=\sum_{m=1}^{j}\sum_{n=1}^{k}F(m,n)H(j-m+1,k-n+1) \qquad (2\text{-}1)$$

where G(j,k) is the $M_1 x M_2$ output array, F(j,k) is the $N_1 x N_2$ input array, and H(j,k) is the $L_1 x L_2$ convolution kernel array, called an impulse response. The input and output dimensions are related by $M_i = N_i + L_i - 1$ for i=1,2. In Eq. (2-1), the symbol $\circledast\circledast$ denotes a two-dimensional convolution. The symbol $\circledast$ will be used to represent a

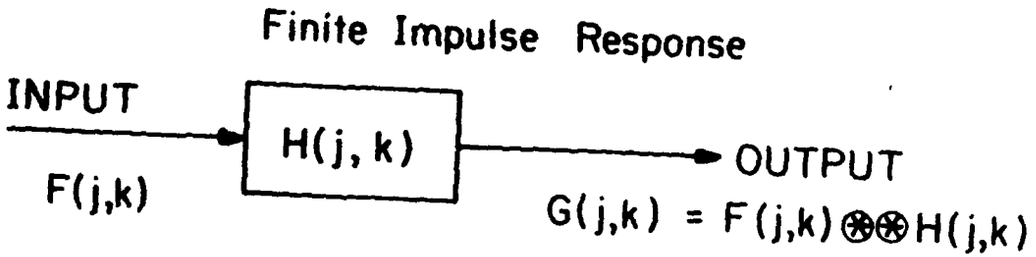


Figure 2-1. The response of a linear system

one-dimensional convolution throughout this dissertation.

In the direct convolution algorithm, the output, G(j,k), is the weighted sum of all values of the input array. The drawback of using the direct convolution algorithm of Eq. (2-1) is that it requires many arithmetic operations. The number of additions and multiplications for direct convolution is $N_1N_2L_1L_2$.

In 1965, Turkey and Cooley [2-3] opened a new era in digital signal processing. They discovered a fast Fourier transform (FFT) algorithm, which is an efficient method for computing a discrete version of the Fourier transform (DFT). The two-dimensional DFT pair of a finite array X(j,k) for j,k = 0,1,...,N-1 can be written in the form

$$\mathcal{X}(u,v) = \frac{1}{N^2}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1} X(j,k)\exp\left\{-\frac{2\pi i}{N}(uj+vk)\right\}$$

$$X(j,k) = \sum_{u=0}^{N-1}\sum_{v=0}^{N-1} \mathcal{X}(u,v)\exp\left\{\frac{2\pi i}{N}(uj+vk)\right\} \qquad (2\text{-}2)$$

where $i = \sqrt{-1}$, u,v are spatial frequency variables, and $\mathcal{X}(u,v)$ denotes the Fourier transform. Both $\mathcal{X}(u,v)$ and X(j,k) are, in general, complex series. Consider the following relation in the frequency domain

$$\mathcal{G}(u,v) = \mathcal{F}(u,v)\mathcal{H}(u,v) \qquad (2\text{-}3)$$

where $\mathcal{G}(u,v)$, $\mathcal{F}(u,v)$ and $\mathcal{H}(u,v)$ are discrete Fourier transform of the array G(j,k), F(j,k), and H(j,k), respectively. By the definition of the DFT, G(j,k) can be expressed as

$$G(j,k)=\sum_{u=0}^{N_1-1}\sum_{v=0}^{N_2-1}\left[\mathcal{F}(u,v)\mathcal{H}(u,v)\right]\exp\left\{2\pi i\left(\frac{uj}{N_1}+\frac{vk}{N_2}\right)\right\} \qquad (2\text{-}4)$$

Thus, computation of the discrete convolution of two arrays can be obtained indirectly using the DFT. Considerable computational efficiency can be gained by the FFT convolution technique. In general, computation requires $N^2+4N^2\log_2 N$ operations when $N_1=N_2=N$ [2-4].

Fourier domain processing is more computationally efficient than the direct convolution of Eq. (2-1) if the size of the impulse response is sufficiently large. The cross over point for the two implementations occurs for a 10x10 impulse response with large input arrays [2-5]. Because, in many practical applications, the size of an impulse response is larger than 10x10, then Fourier domain processing is an efficient computation technique. Furthermore, the efficiency of Fourier domain processing can be increased by overlap-add or overlap-save techniques [2-6].

Several other techniques, for example, number theoretic transforms, have been reported concerning

convolution computation [2-7,2-8]. So far, the techniques we have discussed can be implemented by programs for a general-purpose computer or special-purpose hardware. Recently, due to the dramatic development in Large Scale Integrated (LSI) circuit technology, real-time low cost hardware implementation of a two-dimensional convolution is of great interest. Low cost hardware implementation is possible if the size of the convolution kernels is kept small because the cost of hardware is proportional to the size of the convolution kernel. The technique, commonly referred to as SGK convolution, makes this task possible. A review of these methods is given in Section 2-2. The basic concepts of the SVD technique dealing with nonseparable impulse response and application to sequential convolution is discussed in Section 2-3. A new convolution technique is proposed in Section 2-4. Its application to an image processing display system is described in Section 2-5.

2.2 Review of Small Generating Kernel Convolution

SGK convolution is a processing technique for performing convolution on two-dimensional data arrays by sequentially convolving the arrays with small size convolution kernels. The output of the SGK convolution operation closely approximates the output obtained by convolution with a large kernel prototype filter. The

motivation behind SGK convolution is that it can be used to approximate any impulse response of an FIR filter, and that its structure permits implementation of the convolution by sequential convolution with small size kernels.

McClellan [2-9] was the first to propose a technique for designing such a class of filters by transforming one-dimensional linear phase filters* into two-dimensional linear phase filters. By assuming that the prototype filter is a linear phase filter, his algorithm transforms a one-dimensional filter $h(u)$ into a two-dimensional filter $\mathcal{H}(u,v)$ by means of transformation given by

$$\cos\omega = A\cos u + B\cos v + C\cos u \cdot \cos v + D \qquad (2\text{-}5)$$

The McClellan transformation is an extremely useful tool, requiring only moderate computation, for designing many common types of two-dimensional FIR filters. FIR filters up to order 100 can be designed using this method.

Mersereau et al. [2-10] generalized the McClellan transformation for two-dimensional FIR filters and showed an efficient way to implement the filters designed by this method. The significance of their implementation of the designed filter is that a large two-dimensional convolution

---

*A linear phase filter implies symmetry of the filter.

can be replaced by a sequential convolution with small size kernel operators. A description of the algorithm follows.

The frequency response of a one-dimensional linear phase filter of odd length L is

$$h(u) = \sum_{n=0}^{\frac{L-1}{2}} h(n)\,[\cos u]^n \qquad (2\text{-}6)$$

where h(n) represents the filter impulse response. Because the frequency response of a cascade form is the product of the frequency response of individual stages, the term $[\cos u]^n$ of Eq. (2-6) can be considered as a total frequency response obtained by cascading n identical filters each with a frequency response cos u. It is beneficial to rewrite Eq. (2-6) in terms of the z-transform to obtain

$$H(z) = \sum_{n=0}^{L-1} h(n) z^{-n} = h(0) + \sum_{n=0}^{\frac{L-1}{2}} h(n)\,[p_1(z)]^n \qquad (2\text{-}7a)$$

where

$$p_1(z) = \frac{z+z^{-1}}{2} \qquad (2\text{-}7b)$$

Figures 2-2 to 2-4 show three basic implementation structures proposed by Mersereau et al. [2-11].

Referring to Fig. 2-2, implementation of a two-dimensional filter consists of a $Q(=\frac{L-1}{2})$ stage identical sequential convolution. Note that $p_1(z)$ is replaced by a two-dimensional filter $H_f(z_1,z_2)$, which is obtained by the McClellan transformation. The q-th stage output $O_q(z_1,z_2)$ is obtained from the cumulative sum of the q-th stage as

$$O_q(z_1,z_2) = O_{q-1}(z_1,z_2)+h(q)A_q(z_1,z_2) \tag{2-8a}$$

where

$$A_q(z_1,z_2) = A_{q-1}(z_1,z_2)H_f(z_1,z_2) \tag{2-8b}$$

The term $O_Q(z_1,z_2)$ corresponds to the output array $G(z_1,z_2)$, or equivalently

$$O_Q(z_1,z_2) = \sum_{\ell=1}^{Q} h(\ell)[H_f(z_1,z_2)]^{\ell}F(z_1,z_2) \tag{2-9}$$

The convolution indicated in Eqs. (2-8) and (2-9) could be implemented directly from the direct convolution algorithm of Eq. (2-1). The other structures, shown in Figures 2-3 and 2-4, also can be implemented in a similar manner [2-11]. Mersereau et al. also pointed out that the computational efficiency, i.e., number of multiplication and addition operations required for implementation, is greater than the number for either the direct

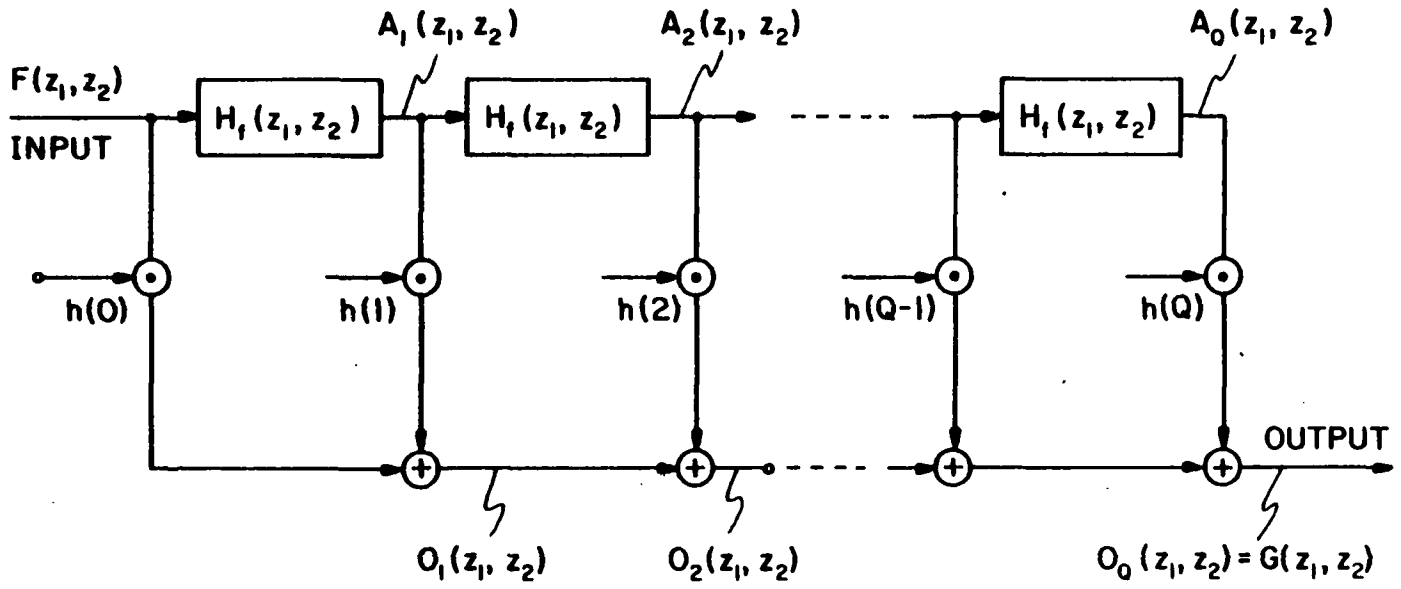


Figure 2-2. Direct transformed implementation

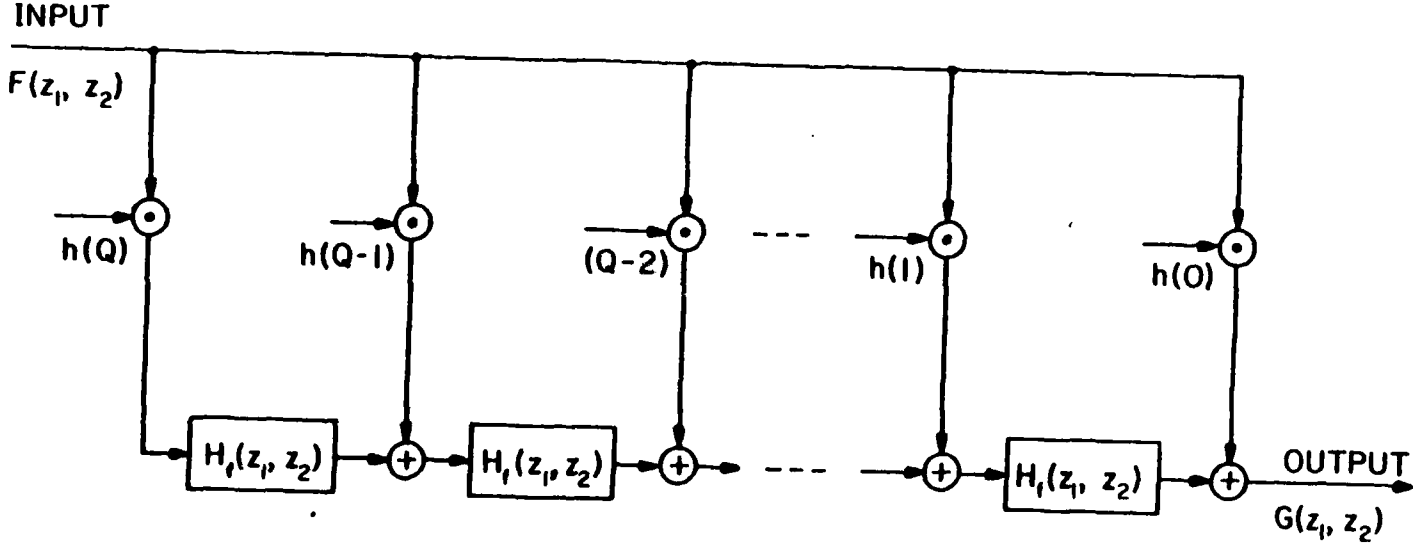


Figure 2-3. Transpose direct transformed implementation

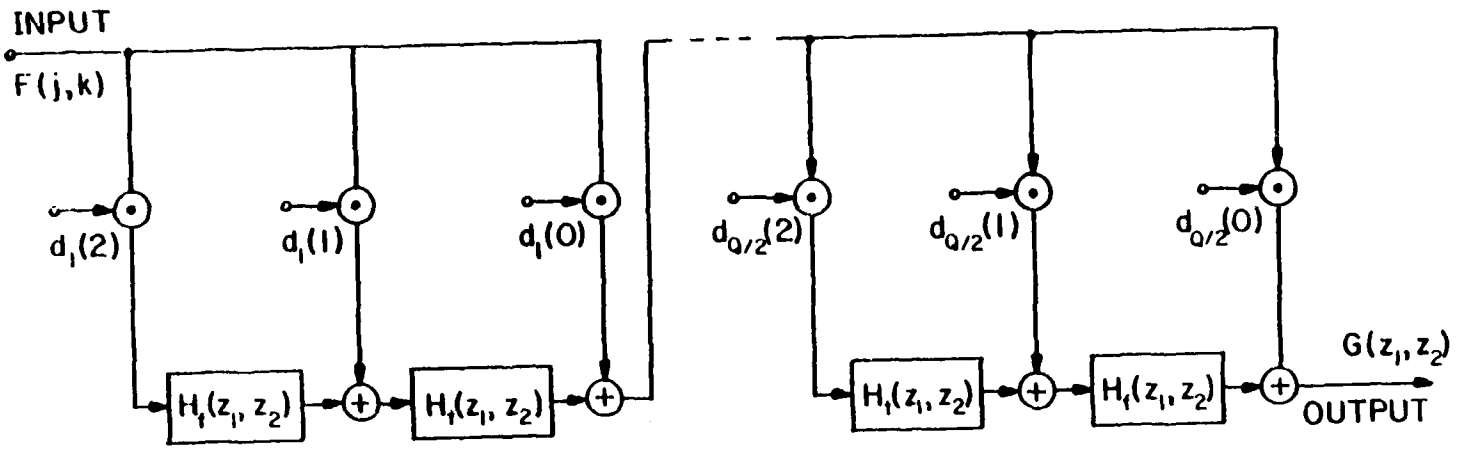


Figure 2-4. Cascade transformed implementation

implementation or an FFT implementation for filters of order up to 50x50.

Another structure of interest, shown in Fig. 2-5, was proposed by McClellan and Chan [2-12]. They noted that $\frac{z^n+z^{-n}}{2}$ is the n-th degree Chebyshev polynomial in the variable $p_1(z)$ of Eq. (2-6). Unfortunately, an arbitrary two-dimensional impulse response cannot be implemented in this way because it is not always symmetrical. The implementations discussed so far are applicable only to McClellan transformed filters. The elementary filters of Figures 2-2 to 2-4 do not necessarily has the same frequency response. The limitation of the previous implementation has motivated a search for more general design techniques for a class of two-dimensional FIR filters that can be easily implemented by sequential convolution with small size kernels, say 3x3.

Abramatic and Faugeras [2-13 to 2-15] presented a synthesis procedure, described in Fig. 2-6, for designing such a class of filters. In comparison with Fig. 2-2, the elementary second-order filters have different transfer functions. The sequential filter proposed by Mersereau et al. is a special case of this class of filters. The design procedure approximates the prototype filter by means of minimizing the mean square error [2-13] or Chebyshev error [2-14] between the approximate and prototype filters.

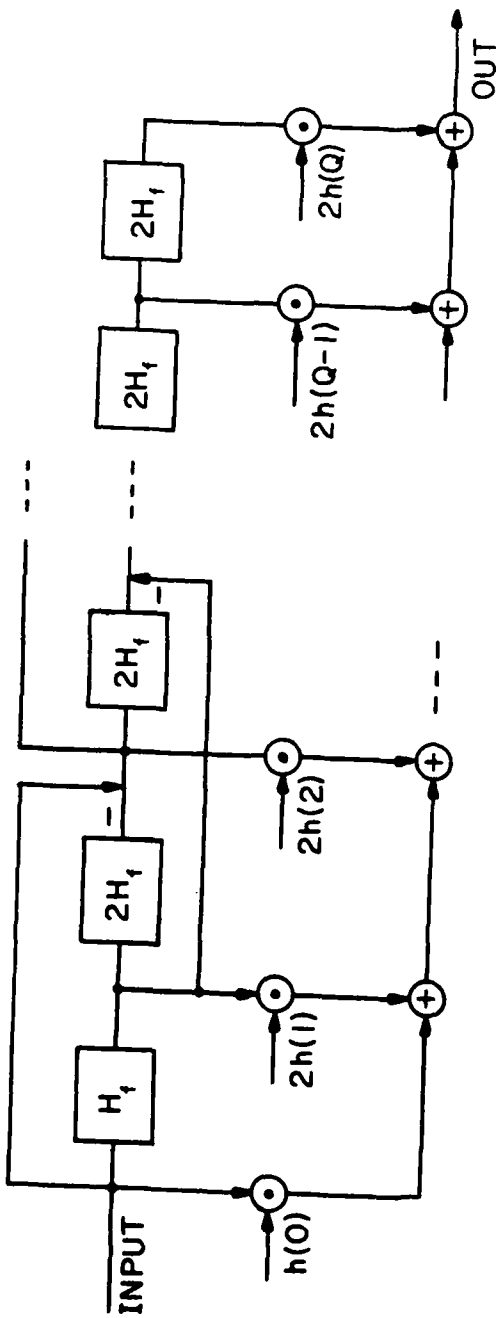


Figure 2-5. Chebyshev structure implementation

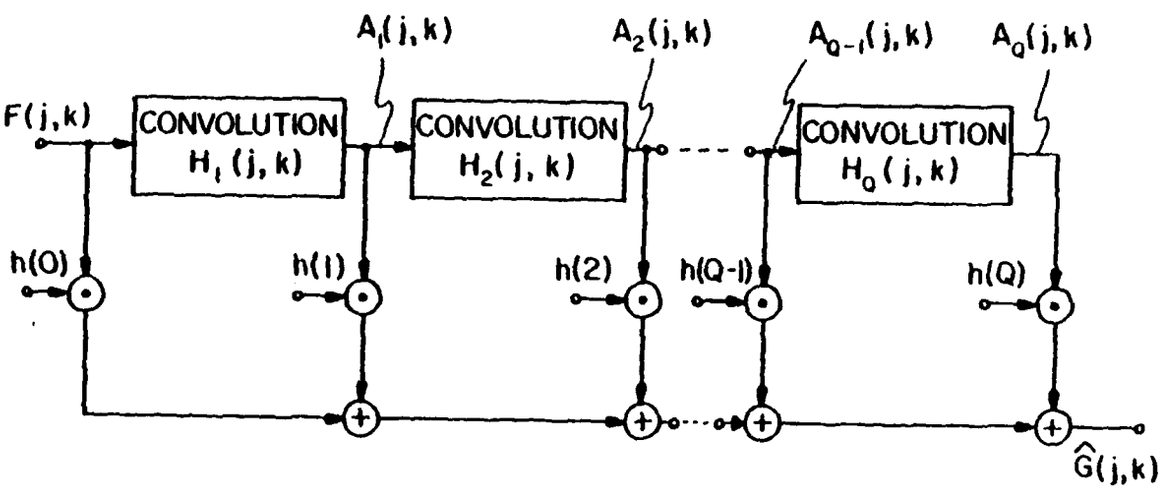


Figure 2-6a. Cumulative SGK convolution system - top ladder configuration

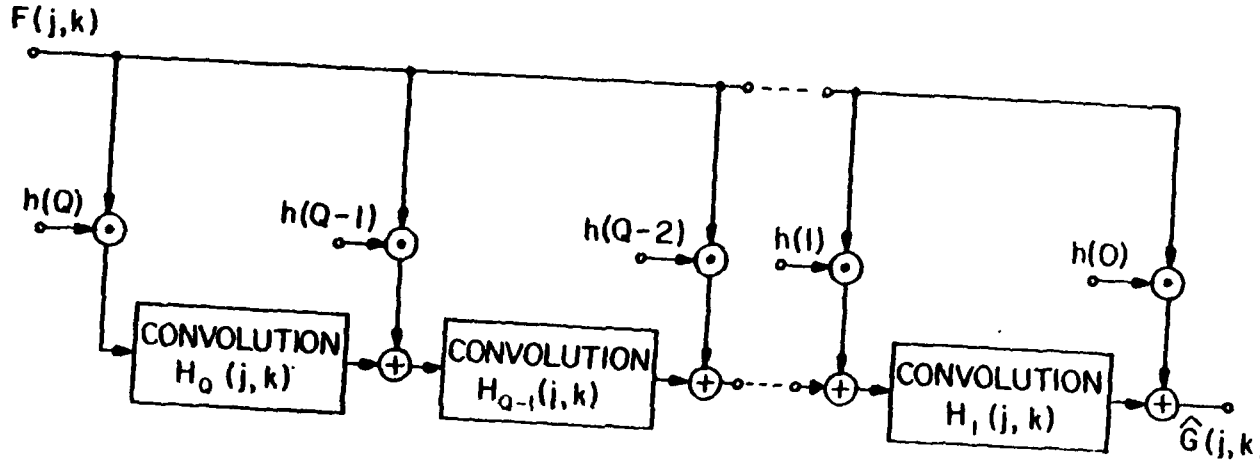


Figure 2-6b. Cumulative SGK convolution system - bottom ladder configuration

Another simple filter with properties similar to those mentioned above is shown in Fig. 2-7.

2.3 SVD Expansion of a Non-Separable Impulse Response Matrix

In the previous sections, an SGK convolution techniques for two-dimensional convolution were discussed. The concern here is with another filter structure based upon SGK convolution with small size kernels, typically 3x1. The basis for the new approach is a matrix expansion by use of the singular value decomposition [2-16]. The reason for choosing the SVD technique in image processing applications is discussed.

A two-dimensional impulse response can be characterized as a matrix. If we consider an arbitrary real impulse response which is modeled by the $L_1 x L_2$ matrix

$$\underline{H} = \begin{bmatrix} H(1,1) & H(1,2) \cdots\cdots\cdots & H(1,L_2) \\ H(2,1) & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ \vdots & & \vdots \\ H(L_1,1) & & H(L_1,L_2) \end{bmatrix} \qquad (2\text{-}10)$$

Suppose the impulse response is spatially invariant and is of seperable form such that

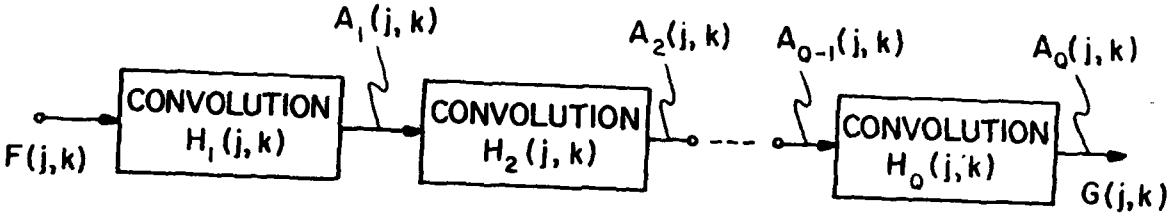


Figure 2-7. Cascade SGK convolution system

$$\underline{H} = \underline{c} \cdot \underline{r}^T \tag{2-11}$$

where $\underline{c}$ and $\underline{r}$ are column vectors representing column and row one-dimensional impulse responses, respectively. We have used the superscript T to denote transposition. Then, two-dimensional convolution may be performed by sequential row and column one-dimensional convolutions. As a result, one can obtain a substantial decease in the number of multiplication and addition operations if the input array size becomes large. If the input array size is NxN, the separable convolution operators of Eq. (2-11) reauires $N^2(L_1+L_2)$ multiplications compared with $N^2 L_1 L_2$ multiplications required in the nonseparable case (fewer are required if the impulse response matrix possesses symmetry). Unfortunately, we cannot assume that the prototype impulse response matrix $\underline{H}$ is always separable. One way of dealing with the nonseparability problem is to use the SVD technique. In the SVD matrix expansion, any arbitrary $L_1 x L_2$ matrix of rank R can be decomposed into the sum of a weighted set of unit rank $L_1 x L_2$ matrices. The significance of the SVD expansion is demonstrated by noting that the nonseparable matrix $\underline{H}$ is the sum of individual separable matrices of unit rank [2-17].

According to the SVD expansion, there exist an $L_1 x L_2$ unitary matrix $\underline{U}$ and an $L_2 x L_1$ unitary matrix $\underline{V}$ for which

$$\underline{U}^T \underline{H}\ \underline{V} = \underline{\Lambda}^{\frac{1}{2}} \tag{2-12}$$

where

$$\underline{\Lambda}^{\frac{1}{2}} = \begin{bmatrix} \lambda^{\frac{1}{2}}(1) & & 0 & \vdots & 0 \\ & \lambda^{\frac{1}{2}}(2) & & & \\ & & \ddots & & \\ 0 & & & \lambda^{\frac{1}{2}}(R) & \\ & & & \vdots & \\ 0 & & & & 0 \end{bmatrix} \qquad (2\text{-}13)$$

is an $L_2 x L_1$ matrix with a general diagonal entry $\lambda^{\frac{1}{2}}(j)$ for j=1,2,...R called a singular value of $\underline{H}$. The singular values can be obtained by square rooting the eigenvalues of $\underline{H}\underline{H}^T$ or $\underline{H}^T\underline{H}$. The columns of $\underline{U}$ are the eigenvectors of $\underline{H}\underline{H}^T$ and the columns of $\underline{V}$ are the eigenvectors of $\underline{H}^T\underline{H}$. Since $\underline{H}\underline{H}^T$ and $\underline{H}^T\underline{H}$ are symmetrical and square, the eigenvalues $\lambda^{\frac{1}{2}}(j)$ are real, and the eigenvectors set $\{\underline{u}_j\}, \{\underline{v}_j\}$ for j = 1,2,...R are orthogonal.

Since matrices $\underline{U}$ and $\underline{V}$ are unitary matrices, Eq. (2-12) is equivalent to Eq. (2-14). Hence, $\underline{H}$ can be decomposed as

$$\underline{H} = [\underline{u}_1\ \underline{u}_2 \cdots \underline{u}_{L_1}]\underline{\Lambda}^{\frac{1}{2}} \begin{bmatrix} \underline{v}_1 \\ \underline{v}_2 \\ \vdots \\ \underline{v}_{L_2} \end{bmatrix} \qquad (2\text{-}14)$$

Equation (2-14) can be reformulated into series form as

$$\underline{H} = \sum_{j=1}^{R} \lambda^{\frac{1}{2}}(j)\,\underline{u}_j\,\underline{v}_j^T \tag{2-15}$$

If we let

$$\underline{c}_j = \lambda^{\frac{1}{2}}(j)\,\underline{u}_j \tag{2-16a}$$

$$\underline{r}_j = \underline{v}_j \tag{2-16b}$$

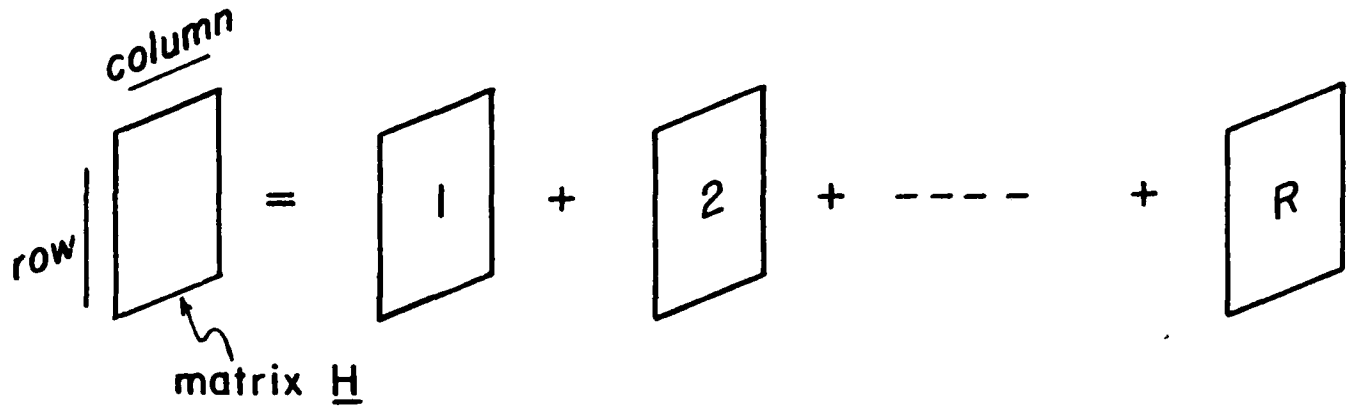
where $\underline{c}_j$ and $\underline{r}_j$ are one-dimensional column and row convolution operators, respectively, then

$$\underline{H} = \sum_{j=1}^{R} \underline{c}_j \cdot \underline{r}_j^T = \sum_{j=1}^{R} \underline{H}_j \tag{2-17a}$$

where

$$\underline{H}_j = \underline{c}_j \cdot \underline{r}_j^T \tag{2-17b}$$

It should be observed that the vector outer product $\underline{u}_j \cdot \underline{v}_j^T$ of the eigenvectors forms a set of separable unit rank matrices each of which is weighted by a corresponding singular value of $\underline{H}$, as shown in Fig. 2-8. If matrix $\underline{H}$ is separable, then we have only one SVD expansion term. If matrix $\underline{H}$ is not separable, theoretically, the exact representation of $\underline{H}$ needs R terms. Hence, the number of multiplication operations for direct convolution requires $RN^2(L_1+L_2)$ multiplication operations, as shown in Fig. 2-9.



$$\underline{H} = \lambda^{1/2}(1)\, u_1 v_1^T + \lambda^{1/2}(2)\, u_2 v_2^T + \text{----} + \lambda^{1/2}(R)\, u_R v_R^T$$

Figure 2-8. SVD expansion into unit rank matrices

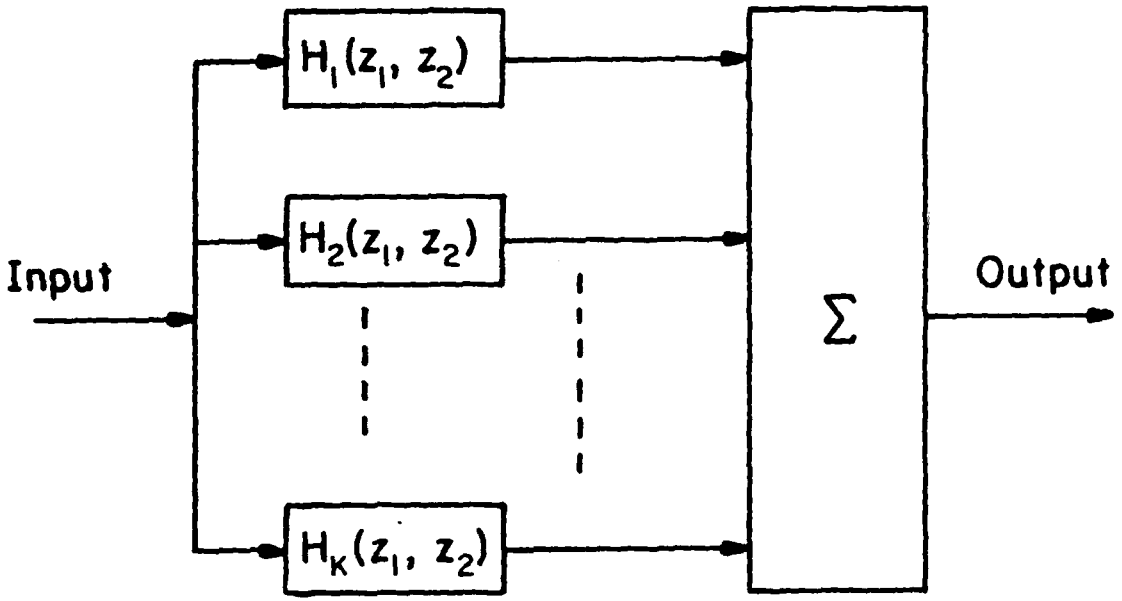


Figure 2-9. Block diagram showing implementation of nonseparable impulse response with SVD expansion

If we assume that the singular values $\lambda^{\frac{1}{2}}(j)$ are listed in order of decreasing magnitude, then the SVD expansion of Eq. (2-17a) can be always rewritten as

$$\underline{H} = \hat{\underline{H}}_{SVD} + \underline{E}_K \tag{2-18a}$$

where

$$\hat{\underline{H}}_{SVD} = \sum_{j=1}^{K} \underline{H}_j \tag{2-18b}$$

$$\underline{E}_K = \sum_{j=K+1}^{R} \underline{H}_j \tag{2-18c}$$

where K is the number of retained term for $\hat{\underline{H}}_{SVD}$ and the "Hat" symbol (^) represents the approximation of $\underline{H}$. The matrix $\underline{E}_K$ denotes the truncation error as a result of retaining the first K terms. Obviously, $\underline{E}_K = \underline{0}$ for K = R. It can be shown that the case for K = 1 corresponds to the minimum mean square error (NMSE) separable approximation of $\underline{H}$ [2-18]. If the elements of $\hat{\underline{H}}_{SVD}$ for K=1 are not close in comparison to the elements of $\underline{H}$, we may add the next largest singular value term for a closer approximation of $\underline{H}$.

In general, we will be satisfied with a multi-stage expansion that will closely approximate $\underline{H}$. One of the most commonly used numerical error measurements is the normalized mean square error (NMSE). Let us define the SVD

approximation error ε_K for the measurement of the degree of approximation by retaining only the first K terms in the expansion as

$$\varepsilon_K = \left[\frac{\sum_i \sum_j |E_K(i,j)|^2}{\sum_i \sum_j |H(i,j)|^2} \right]^{\frac{1}{2}} \qquad (2\text{-}19)$$

If all singular values are the same magnitude, we have to retain R terms. If, however the first few singular values are very large compared with the magnitude of the rest of the singular values, it would be sufficient to retain only the first few terms for approximation. Two questions naturally arise. How many terms will be sufficient for close approximation in most practical cases? What characteristics of impulse responses are required to approximate $\underline{H}$ by a few singular value terms?

In most cases, an imaging system can be modeled by a superposition integral relating the input and output continuous fields of a linear system [2-4]. In order to reduce the problem to a discrete model, the point spread function (PSF) of the imaging system, as well as the input and output images, should be discretized. The matrix $\underline{H}$ resulting from the PSF samples is nearly singular or ill-conditioned since the rows of the matrix $\underline{H}$ are approximately a linear combination of one another [2-19,2-20].

Ill-conditioning of a matrix can be described by its condition number [2-21]. The condition number of given matrix $\underline{A}$ is defined in terms of the ratio

$$C[\underline{A}] = \frac{\lambda_{max}^{\frac{1}{2}}}{\lambda_{min}^{\frac{1}{2}}} \tag{2-20}$$

of the largest $\lambda_{max}^{\frac{1}{2}}$ to smallest $\lambda_{min}^{\frac{1}{2}}$ singular value of $\underline{A}$. The condition number is a useful tool for explaining the effect of perturbation caused by additive noise on the accuracy of computation involved [2-22]. The condition number approaches infinity as $\lambda_{min}^{\frac{1}{2}}$ goes to zero. In this case, the matrix is called ill-conditioned and will have a large condition number. In an ideal imaging system, characterized by a delta function point spread function, the condition number is unity since all singular values have the same magnitude. Sometimes it is convenient to demonstrate matrix conditioning by showing singular value magnitude plots. Referring to Fig. 2-10, a well-conditioned matrix requires more terms in a SVD expansion than an ill-conditioned matrix. But, it is noted here that ill-conditioned and nearly singular problems are very common in imaging systems [2-4]. Therefore, we do not need to retain all terms in the SVD expansion, but only a few terms because of ill-conditioning of the PSF matrix itself. The usefulness of the SVD expansion can be demonstrated by noting that we can trade off between the

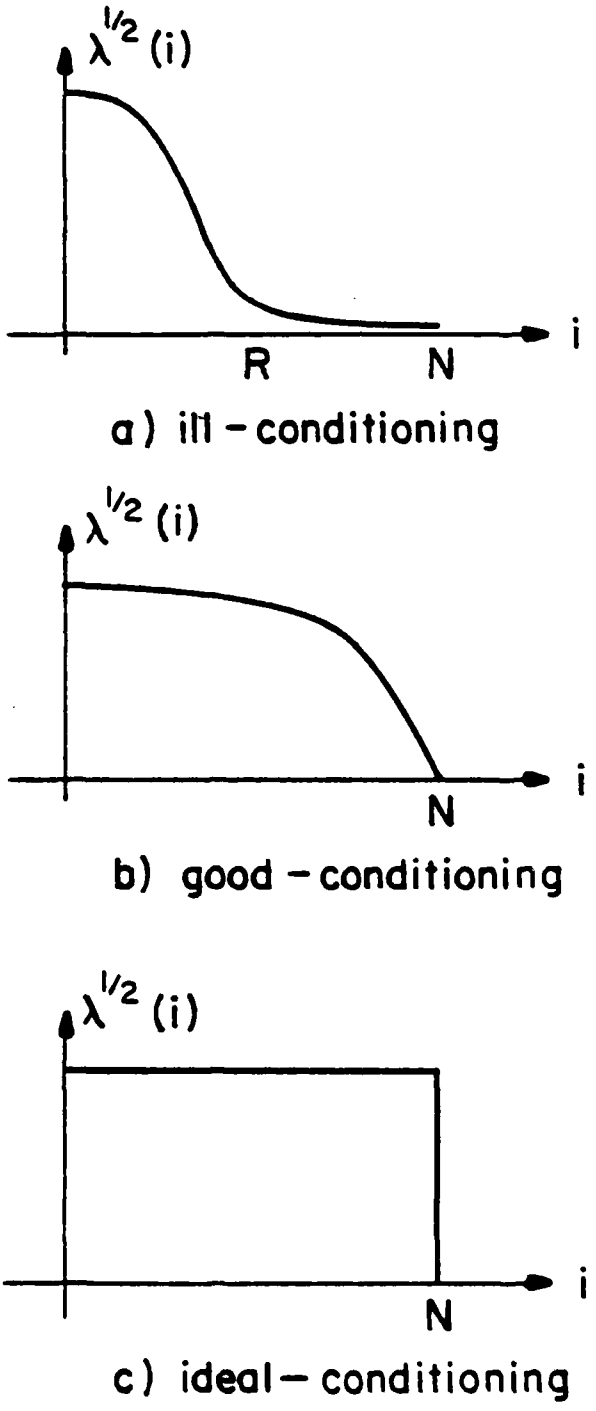


Figure 2-10. Singular value plot

amount of NMSE and the computational efficiency by choosing the number of terms in the SVD expansion. By retaining only K terms in the SVD expansion, the required multiplication is $KN^2(L_1+L_2)$. Computational efficiency still holds as long as $KN^2(L_1+L_2) \leq N^2 L_1 L_2$, or $K \leq \frac{L_1 L_2}{(L_1+L_2)}$.

2.4 The SVD/SGK Cascade Convolution Technique

In the previous section, approximion of a nonseparable impulse response matrix $\underline{H}$ in terms of the sum of individual separable matrices of unit rank was discussed. To implement the SVD convolution, each separable convolution operator is implemented in parallel, and summed together, as shown in Fig. 2-9. In this section, SVD and SGK techniques are combined to obtain a more versatile two-dimensional convolution technique requiring a simpler implementation.

Since each SVD expanded separable matrix of unit rank is an outer product of the one-dimensional column and the row operator $\underline{c}_j$ and $\underline{r}_j$, here each $\underline{c}_j$ and $\underline{r}_j$ is to be considered as a one-dimensional FIR filter. There are a variety of alternative forms in the FIR filter realization. Realization of FIR filters generally takes the form of a nonrecursive computation algorithm. One way of realizing FIR filters for hardware simplicity is to use a cascade form. In the cascade form, the z-transform of the impulse response with the length of L can be expressed as a product

of second-order SGK filters as

$$H(z) = \prod_{k=1}^{Q} H_k(z) = \prod_{k=1}^{Q} [\beta_{o,k} + \beta_{1,k} z^{-1} + \beta_{2,k} z^{-2}] \qquad (2\text{-}21)$$

where the $\beta_{j,k}$ are real numbers and Q, the number of convolution stages, is

$$Q = \begin{cases} \frac{L-1}{2} & , \text{ L odd} \\ \frac{L}{2} & , \text{ L even} \end{cases} \qquad (2\text{-}22)$$

When L is even, one of the kernel terms $\beta_{2,k}$ will be zero. Here we shall be concerned only with the case of odd length impulse response. The kernel of each second-order SGK filter can be easily obtained by solving the zeros of the polynomial H(z) because H(z) is a polynomial in z^{-1} of degree L-1.

A new approach for two-dimensional SVD/SGK convolution, shown in Fig. 2-11, is to realize each one-dimensional convolution operator $\underline{c}_\ell$ and $\underline{r}_\ell$ for $\ell = 1,2,\ldots,K$ as a sequence of second-order SGK filters. Referring to Fig. 2-11, the z-transform of the SVD/SGK convolution filter is

$$\hat{H}(z_1,z_2) = \sum_{\ell=1}^{K} C_\ell(z_1) R_\ell(z_2) \qquad (2\text{-}23a)$$

or

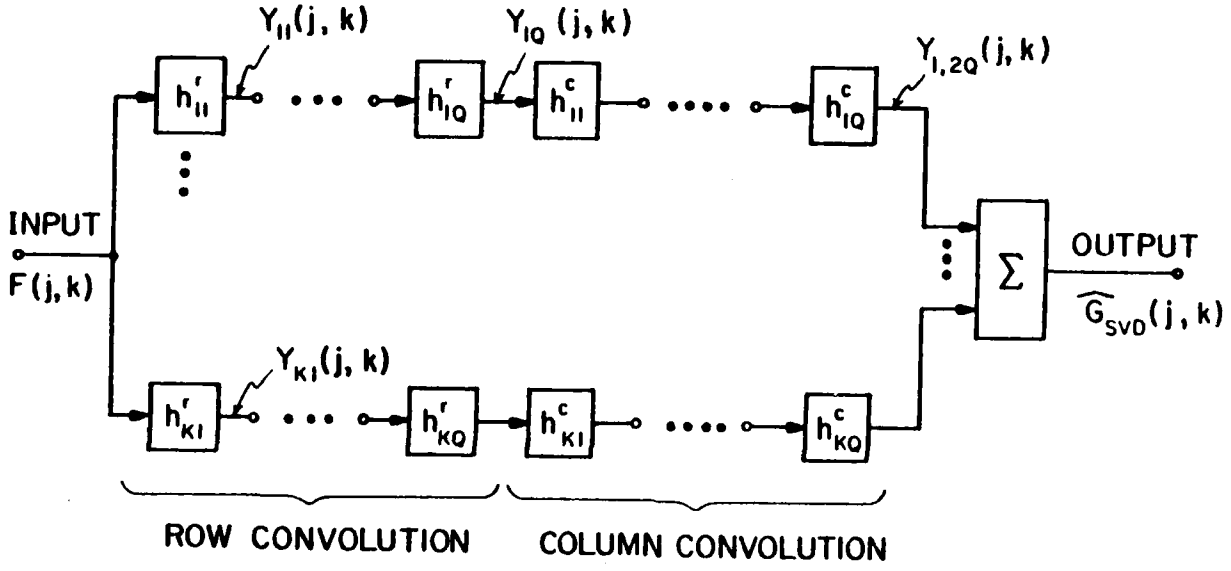


Figure 2-11. SVD/SGK convolution system

$$H(z_1,z_2)=\sum_{\ell=1}^{K}\left[\prod_{i=1}^{Q}C_{\ell,i}(z_1)\right]\left[\prod_{j=1}^{Q}R_{\ell,j}(z_2)\right] \qquad (2\text{-}23b)$$

where

$$C_{\ell}(z_1) = \prod_{i=1}^{Q}C_{\ell,i}(z_1) \qquad (2\text{-}24a)$$

$$R_{\ell}(z_2) = \prod_{i=1}^{Q}R_{\ell,j}(z_2) \qquad (2\text{-}24b)$$

The terms $C_{\ell}(z_1)$ and $R_{\ell}(z_2)$ for $\ell=1,2,\ldots K$ denote the z-transform of each column and row one dimensional convolution operator, as defined in Eq. (2-16), and each $C_{\ell,i}(z_1)$, $R_{\ell,i}(z_2)$ for $i,j = 1,\ldots Q$ is the z-transform of the second-order SGK filter.

One of the most important reasons for using FIR filters is that they can be designed to possess linear phase, a feature that is very useful in speech processing and data transmission. It is easy to see where the zeros of such linear phase FIR filters can lie by examining their z-transforms because a linear phase filter is symmetrical. In the general case, the filter system function is

$$H(z) = \sum_{n=0}^{L-1}h(n)z^{-n} \qquad (2\text{-}25)$$

Linear phase FIR filters have a symmetry property such that

$$h(n) = h(L-1+n) \tag{2-26}$$

Therefore, by using Eq. (2-26), Eq. (2-25) can be rewritten as

$$H(z) = z^{-\frac{(L-1)}{2}} \left\{ h(0) \left[z^{\frac{L-1}{2}} + z^{-\frac{(L-1)}{2}} \right] + h(1) \left[z^{\frac{L-3}{2}} + z^{-\frac{(L-3)}{2}} \right] + \ldots \right\} \tag{2-27}$$

If z is replaced by z^{-1}, then we obtain

$$H(z) = z^{\frac{(L-1)}{2}} \left\{ h(0) \left[z^{-\frac{(L-1)}{2}} + z^{\frac{(L-1)}{2}} \right] + h(1) \left[z^{-\frac{(L-3)}{2}} + z^{\frac{(L-3)}{2}} \right] + \ldots \right\} \tag{2-28}$$

By comparing Eq. (2-27) with Eq. (2-28), one obtains

$$H(z) = z^{-(L-1)} H(z^{-1}) \tag{2-29}$$

Equation (2-27) shows that the zeros of H(z) are identical to the zeros of $H(z^{-1})$. In other words, if H(z) has a complex zero a+ib, with $a^2+b^2 \neq 1$, then H(z) must have a minor image zero $\frac{1}{a+ib}$. Since the impulse response of the filter is a real number, every complex zero of H(z) has its complex conjugate as another zero.

The discussion above leads immediately to the possible form of $H_k(z)$. For every complex zero of H(z), $a^2+b^2 \neq 1$,

the second-order SGK filter will be of the form

$$H_k(z) = [z^{-1}-(a_k+ib_k)][z^{-1}-(a_k-ib_k)] \qquad (2\text{-}30)$$

If the zero, a , is not complex, then the form of SGK filter is

$$H_k(z) = [z^{-1}-a][z^{-1}-\frac{1}{a}] \qquad (2\text{-}31)$$

If the zeros are either -1 or 1, then the zero is its own complex conjugate as well as a mirror image. In this case, the form of the SGK filter is

$$H_k(z) = (z^{-1}\pm 1) \qquad (2\text{-}32)$$

From the discussion above, the following rule of zero grouping can be stated:

1) Complex zeros are grouped together in conjugate pairs.
2) Real zeros, that are reciprocal of each other, are paired together.
3) Double or higher multiplicity zeros are paired together in pairs.

The rule of zero grouping guarantees that all kernels are real numbers. The proposed SVD/SGK convolution has both advantages and disadvantages. Since two-dimensional large kernel convolution is replaced by a cascade of one-dimensional SGK filters, the processing complexity can

be reduced. Also, from a theoretical point of view, there is no approximation error in realizing the cascade form because all kernels can be found exactly by simply solving for the zeros of $H(z)$. Only the SVD truncation error defined in Section 2-3 will be introduced. On the other hand, computational inefficiency could be one of the disadvantages of replacing two-dimensional SGK filters by one-dimensional SGK filters. It is possible, however, to perform two-dimensional SGK filter convolution instead of one-dimensional since we can rewrite Eq. (2-23) in the alternative form

$$\hat{H}(z_1,z_2) = \sum_{\ell=1}^{K}\left[\prod_{i=1}^{Q} H_{\ell,i}(z_1,z_2)\right] \qquad (2\text{-}33a)$$

where

$$H_{\ell,i}(z_1,z_2) = C_{\ell,i}(z_1,z_2)R_{\ell,i}(z_1,z_2) \qquad (2\text{-}33b)$$

As a matter of fact, the two-dimensional SGK filter will increase computational speed by a factor of two, but the hardware is more costly and the processing more complex. Implementation of a two-dimensional SGK filter, in general, requires nine multipliers and adders.

2.5 Image Processing Display Implementation

There are many ways to implement the SVD/SGK convolution method. The goal of this section is to

describe how to organize the implementation and apply SVD/SGK convolution to an image processing display system. Let us denote F(j,k) as a filter input array with a size of NxN and the array G(j,k) as its output. We also assume that the size of the prototype impulse response is (2Q+1)x(2Q+1). For simplicity, we shall discuss only implementation for one term in the SVD expansion because the SVD/SGK convolution consists of K identical paths. The implementation iterates 2Q stages. The node labeled $Y_i(j,k)$ for i = 1,2,...,2Q is an intermediate array, which will be used in the next convolution. In other words, at each node, the array $Y_{i-1}(j,k)$ is used to produce array $Y_i(j,k)$. Therefore, $Y_{2Q}(j,k)$ corresponds to the final output array G(j,k). Once the array $Y_i(j,k)$ has been computed from $Y_{i-1}(j,k)$, $Y_{i-1}(j,k)$ is no longer needed and that Y_i can then be stored in place of Y_{i-1}. To implement SVD/SGK convolution, it is then necessary to have at least one common storage for the intermediate array $Y_i(j,k)$ for i = 1,2,...,2Q. But the storage array should be initialized by the input array F(j,k). As the computations proceed along the chain of SGK filters, each $Y_i(j,k)$ will be larger in extent than its predecessor $Y_{i-1}(j,k)$. Therefore, the required storage size must be large enough to hold (N+2Q)x(N+2Q) pixels.

Because the implementation of SVD/SGK convolution is highly modular, the concept of SVD/SGK convolution is

ideally suited for implementation by a digital image display system. Two-dimensional convolution performed by a digital computer in image processing is often quite time consuming because of the serial nature of the computation and the slow input-output transfer rate between the computer and display [2-23]. But solid state device technology makes it possible to develop memory devices that produce pixels at a serial rate of about 10 million per second. Figure 2-12 is a basic diagram of the architecture for SVD/SGK convolution [2-23]. In the operation of this hardware, an original image to be convolved is written into an accumulator memory with a size of (N+2Q)x(N+2Q). The accumulator will thus appear as an array of nonzero values encircled with Q square rings of zeros. Then the input array is sequentially convolved with a 3x1 impulse response operator, depending on the row or column direction. Three multiplication and three addition operations are performed for each pixel. After each convolution, the microprocessor will update the kernels of the 3x1 convolution operator. This process proceeds for 2Q stages, equivelent to 2Q frame time. Thus, after 2Q frame times, the contents of the accumulator memory are added to the partial sum memory, which is initialized by zero, and return to the original image. This processing completes the first term of the SVD expansion. The partial sum memory can be displayed, if desired. This process is repeated for the remaining SVD

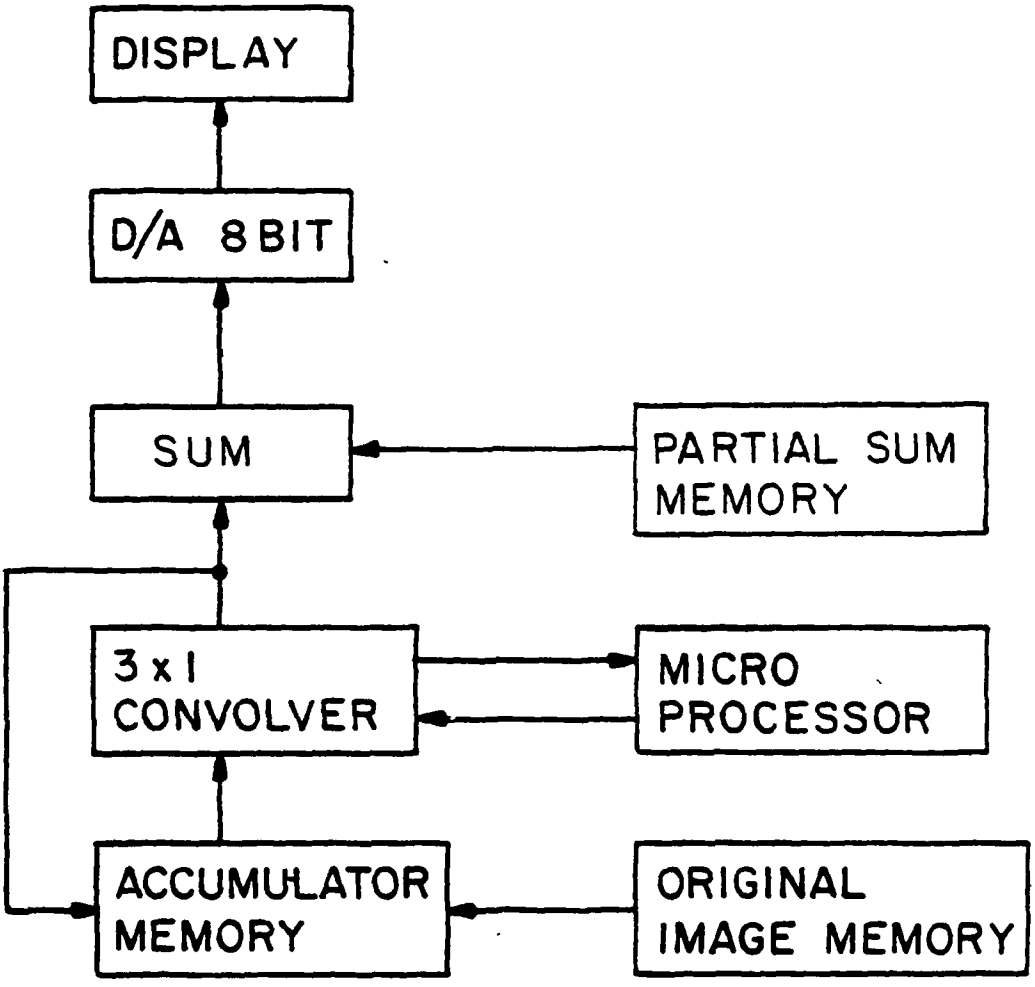


Figure 2-12. SVD/SGK convolution architecture

terms, resulting in a total processing time of 2KQ frame time intervals. For conventional 30 frame/second operation, the SVD/SGK convolution operation can be completed in 2KQ/30 seconds, far less than the 20 to 30 seconds required by a hardware floating point processor.

2.6 Conclusion

In this chapter, it was shown that the SVD expansion of the impulse response of a two-dimensional FIR filter is a very useful technique for a two-dimensional convolution. The SGK and SVD/SGK convolution methods are attractive techniques for simplifying the computational requirement of two-dimensional convolution. The SVD/SGK convolution approach is attractive for two reasons. First, large two-dimensional convolution is replaced by sequential one-dimensional convolution with small size convolution operators. If one is interested in implementing SVD/SGK convolution with special-purpose hardware, that approach reduces both the cost and the complexity of the processing. Second, the design for the SVD/SGK convolution filter is simple and fast, and the design procedure introduces a very small approximation error caused by retaining only the first few terms of the SVD expansion. But the design for the SGK convolution filter generally leads to complicated, time-consuming nonlinear optimization programs. To utilize SVD/SGK convolution on the digital image display system, a

basic diagram of the architecture for SVD/SGK convolution was introduced.

CHAPTER 3

FINITE WORD-LENGTH EFFECTS IN SVD/SGK CONVOLUTION

3.1 Introduction

Until now, we have assumed full precision implementation for SVD/SGK convolution. We will now discuss some practical problems that must be considered when digital signal processing algorithms are implemented with programs for general-purpose computers or, especially, with special-purpose hardware. These problems are caused by the use of finite word-length registers to represent signal values, coefficient values, and arithmetic operations. Because of finite word-length, a quantized number will not take the exact value assigned by the design procedure.

When a signal, to be processed digitally, is obtained by sampling a band-limited signal, the numbers must be represented by a finite word-length register in the digital machine. This conversion process between analog samples and discrete valued samples is called the quantization process. This quantization process is an irreversible nonlinear operation. When the filter coefficients are

quantized for digital implementation, the resulting filter must be checked to be sure that it is close enough to the desired response. In addition, finite word-length operation has a strong effect on both the cost and speed of the system. If the word-length is large, then the cost of hardware will be expensive and the processing speed low. Therefore, reducing the word-length as much as possible is a major goal.

It should be noted here that effects of finite word-length in a digital filter depend on many issues such as whether fixed-point or floating-point arithmetic is used, whether the fixed-point number represents a fraction or an integer, and whether quantization is performed by rounding or truncating.

In a digital system, numbers, generally, are represented by a radix of 2. Thus, numbers are represented by strings of binary digits, either zero or one. If a word-length of b bits is chosen to represent numbers, 2^b different numbers can be represented. There are two ways to represent binary numbers, depending on the location of binary points. In fixed-point arithmetic, the position of a binary point is assumed to be fixed. The bits to the right of a binary point are the fractional part and those to the left of the binary point are the integer part. But, with no loss of generality, we assume throughout this

dissertation that the position of the binary point is just to the right of the first bit. Thus, the range of numbers that can be represented with b bits is -1.0 to $1.0-2^{-(b-1)}$. It is noted that the signal magnitude can be scaled to any desired range. Certainly, the binary point could be moved further to the right to allow a signal with magnitude greater than unity, but the price paid is greater complexity in hardware.

There are three formats commonly used to represent fixed-point numbers, depending on the way of expressing negative numbers: sign and magnitude, 2's complement, 1's complement. The sign and magnitude, the most simple format, represents the magnitude by a binary number; the sign is represented by the leading digit. It is useful to assume that in all three representations, the leading bit is zero for a positive number and one for a negative number. For this reason, the leading bit is called a sign bit. But the sign and magnitude format presents an inherent problem in performing simple arithmetic, such as addition. Therefore, the sign and magnitude format is generally avoided in a digital system.

For 1's complement representation, positive numbers are represented as in the sign and magnitude format. A negative number is represented by complementing all of the bits of the positive number. In 2's complement

representation, positive numbers are represented as in the sign and magnitude format. But a negative number is represented by subtracting the magnitude from 2.0. The choice among the three formats depends on hardware consideration. The two's complement format is widely chosen in most digital systems because it conveniently performs subtraction using an adder. Another advantage of using the 2's complement format is that the correct total sum will be obtained even when partial sums overflow or underflow.

In fixed-point arithmetic, the result of adding two b-bit numbers is still b bits. However, the magnitude of the resulting sum can exceed unity. This phenomenon, called overflow, is inherently related to the limited dynamic range of fixed-point arithmetic. Scaling can be performed to avoid undesired overflow. The product of two b-bit numbers results in a 2·b-bit number. If multiplication is carried out p times, the required word-length for representing the result is p·b bits. This is clearly an unacceptable situation for the hardware and economy. To remedy this problem, truncating or rounding operations to fit the results of multiplication into a finite word-length register is necessary. The error due to truncating or rounding of p bits of word-length into q bits ($p \geq q$) of word-length is commonly referred as roundoff error. Considerable attention has been paid to

investigating the effect of roundoff error on digital filter implementation in the last decade [3-1 to 3-5].

Floating-point arithmetic is a method for providing automatic scaling. An arbitrary finite number x can be represented exactly using the floating point representation

$$x = \text{sign}(x)\, c \cdot 2^{i} \qquad (3\text{-}1)$$

where c, the mantissa, is a full precision binary number such that $1/2 \leq c \leq 1$ and i, the exponent, is an integer. The number of bits, b, in a flowing-point representation should be divided into the number of bits b_1, for the mantissa and the number of bits b_2 for the exponent. Although floating-point arithmetic requires truncating or rounding operations in both multiplication and addition [3-6], it provides more dynamic range than fixed-point arithmetic.

The comparison between fixed-point and floating-point arithmetic depends on the input probability density function, input power spectral density, and filter frequency response [3-7]. If the floating-point mantissa and fixed-point word-length have the same word-length, then floating-point arithmetic is more advantageous. Generally, when a large dynamic range is required, floating-point arithmetic generates less roundoff noise because it provides automatic scaling [3-1]. But it should be noted

here that floating-point arithmetic is significantly more complicated and expensive in hardware than fixed-point arithmetic. When economy and speed are of major concerns, fixed-point arithmetic is usually a logical choice.

The comparison between truncating and rounding depends on whether fixed-point or floating point arithmetic is used and how negative numbers are represented. However, experiments have shown that the errors generated by truncation are more severe than those generated by rounding because of a bias [3-3]. Truncation operation is not commonly used in practical digital system.

The next problem of concern is fraction or integer multiple representation of numbers. In integer multiple representation, all numbers are represented by 2^{-N}, where N is an integer. Therefore, the multiplication operation requires only a shift operation. This shift operation will increase computational speed and simplify the hardware. But one can expect losses in dynamic range and accuracy in arithmetic. Since accuracy is essential in finite word-length arithmetic, fraction representation is commonly chosen.

Due to all these reasons, attention will be focused on fixed-point arithmetic with the rounding operation and fraction representation. Another reason for restricting our attention to fixed-point arithmetic is that the

overflow resulting from the limited dynamic range can be avoided by proper scaling of the signal level.

3.2 Preliminary Statement

Fixed-point arithmetic with finite word-length causes three common error sources [3-3]:

1) Quantization of the input signal into a set of discrete values causes inaccuracies.

2) Representation of the filter coefficients by a finite word-length changes the filter characteristics.

3) Rounding or truncating of the results of arithmetic operations within the filter causes errors, known as roundoff* noise in the filter output.

Overflow can also be a problem within filters. However, the overflow problem can be avoided if the signals are properly scaled. This problem will be discussed later.

The first source of error above, commonly referred to as A/D noise, is inherent in any analog-to-digital (A/D)

---

*This term is universally adopted whether rounding or truncation operation is actually performed.

conversion process, and has been studied in great depth [3-5]. It is noted here that the input data array is already a quantized version in most practical cases. For example, 8-bit image data is common in image processing. Furthermore, it shall be shown later that the effect of input quantization is far less severe than the effect of roundoff noise. Hence, the effect of A/D noise has been excluded in this study.

The second source of error mentioned above occurs because the filter coefficients, following a design procedure, which would normally use full precision, must be quantized with finite word-length. This quantization of the filter coefficients will alter the transfer function. This error differs from structure to structure. It is advantageous to use a structure that is insensitive to filter coefficient quantization. In general, the effect of filter coefficients in accuracy is most severe in a higher-order filter when the filter is realized in the direct form than when it is realized in the parallel or cascade form. As a rule, therefore, the parallel or cascade form should be used for higher-order filters whenever possible [3-3]. Experimental results have shown that the amount of error is not significant in our case. Therefore, no particular emphasis will be made in this study, except in Chaper 5.

The third source of error mentioned above is of major concern in fixed-point arithmetic, and is the major subject of the next section. Roundoff noise is the most important factor in determining the complexity of hardware and speed. Large word-length will slow down computational speed. Furthermore, the price paid by increasing the word-length for filter coefficients is negligible compared to the price paid by increasing the word-length for reducing roundoff error. In addition, a limit cycle can occur in the recursive realization of FIR filters [3-9]. However, a limit cycle cannot occur in the nonrecursive structures.

3.3 Fixed-Point Arithmetic

3.3.1 Roundoff Error

The direct form of discrete convolution can be characterized as a calculation of the sum of products

$$S = \sum_{n=1}^{N} a(n)b(n) = \sum_{n=1}^{N} c(n) \qquad (3\text{-}2)$$

Let us assume that a(n) and b(n) are (b+1)-bit numbers (including sign bit) and products are rounded to less than (2b+1) bits, but more than (b+1) bits. Then, the rounded products can be written as

$$[c(n)]_r = c(n) + e(n) \qquad (3\text{-}3)$$

The relation between $[c(n)]_r$ and $c(n)$ is shown in Fig. 3-1, where $[c(n)]_r$ denotes the rounded number and $e(n)$ represents the error resulting from rounding. In fixed-point arithmetic, the error made by rounding with (b_1+1) bits satisfies the inequality

$$-\frac{2^{-b_1}}{2} \leq e(n) \leq \frac{2^{-b_1}}{2} \tag{3-4}$$

Thus, the resulting sum can be expressed as

$$S_1 = \sum_{n=1}^{N} [c(n)]_r = S + \sum_{n=1}^{N} e(n) \tag{3-5}$$

Let us assume that the resulting sum S will be stored into (b_2+1) bits of word-length. Then, the resulting sum rounded to (b_2+1) bits can be rewritten as

$$S_2 = S_1 + v = S + \sum_{n=1}^{N} e(n) + v \tag{3-6a}$$

where

$$-\frac{2^{-b_2}}{2} \leq v \leq \frac{2^{-b_2}}{2} \tag{3-6b}$$

Therefore, by combining Eqs. (3-5) and (3-6), we obtain

$$|S_2 - S| \leq \frac{2^{-b_1}}{2} N + \frac{2^{-b_2}}{2} \tag{3-7}$$

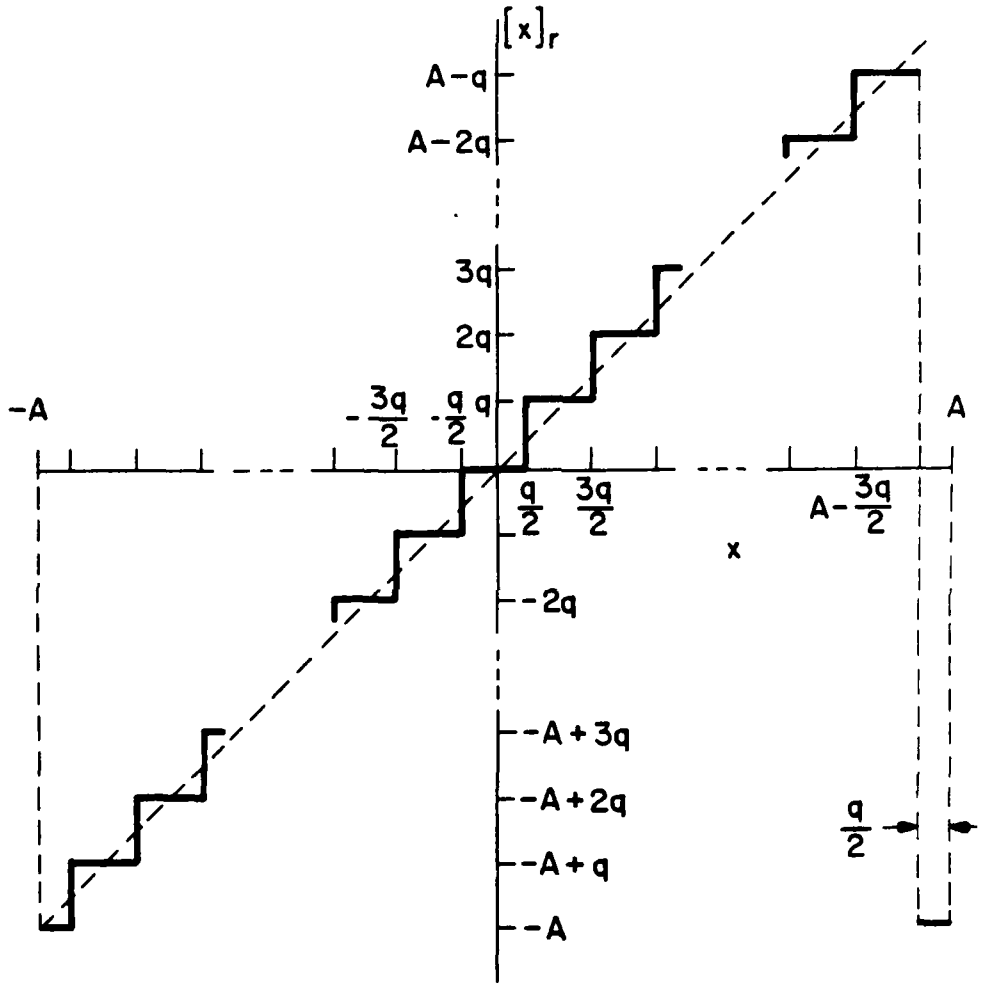


Figure 3-1. Rounding of two's complement number

The characteristic of the roundoff noise at the output depends on the location where rounding is performed. There are two possible locations for rounding. First, if all multiplications are performed with full precision, rounding is performed only after summation. Then, from Eq. (3-5), $e(n) = 0$ for $n = 1,2,\ldots N$ so that

$$|S_2 - S| \leq \frac{2^{-b_2}}{2} \tag{3-8}$$

If all multiplication products are rounded for storage before addition, $v = 0$ and

$$|S_2 - S| \leq \frac{2^{-b_1}}{2} N \tag{3-9}$$

Unfortunately, all of the bounds derived are for worst cases, and thus, are of little practical usefulness. In the following discussion, we will derive more useful bounds.

A less conservative estimate of the noise caused by rounding can be obtained by a statistical approach [3-3]. It should be noted here that a precise analysis of roundoff noise is generally complicated, and not required in practical applications. The purpose of error analysis is to determine word-length within a filter to satisfy some specification with reasonable tolerance. Furthermore, a

final decision concerning word-length is insensitive to inaccuracies in the error analysis. Thus, an analysis correct to within 30 % to 40 % is often acceptable [3-6].

The statistical approach considers the errors introduced by rounding to be small random quantities. This viewpoint simplifies the analysis and enables useful theoretical results to be derived. Many computer simulations results have verifed the validity of the statistical approach [3-3 to 3-5]. It has been claimed that the statistical approach tends to be more accurate when the number of quantization levels is not too small [3-3].

Three common assumptions are made concerning the effect of rounding [3-3]. They are:

1) The error sequence e(n) is a white-noise sequence.

2) The probability distribution of the error sequence e(n) is uniform over the range of quantization intervals.

3) The error sequence e(n) is uncorrelated with the input and itself.

The uncorrelatedness assumption is particularly attractive because the total error due to rounding is the sum of each rounding error. There are some controversies

over the validity of the assumptions we have made. For instance, if the input is constant, we will see clearly that all three assumptions above are invalid. In such cases, the roundoff noise is a deterministic quantity, and is no longer uncorrelated with the input. But these assumptions seem to be valid for most filters with input signals of reasonable amplitude and spectral context. If uncorrelatedness is not assumed, then the analysis will be more complicated, and the results will be dependent on the particular input signal or class of input signals [3-2].

Based on the discussion above, Fig. 3-2 shows a noise model of a 3x1 SGK filter in which the rounding operation is replaced by an additive roundoff noise. In this model, we assumed that all multiplication products are represented exactly, and rounding is performed only after they are summed, i.e., at the filter output. Then only one noise source is present in the filter, and it superimposes on the output.

There are 2Q 3x1 SGK filters in each SVD expansion stage, Q columns and Q rows. Let us define a two-dimensional 3x3 filter, $t_{j,i}^{(\ell,m)}$, depending on the column or row direction. The subscript j denotes the j-th stage SVD expansion. Thus, let

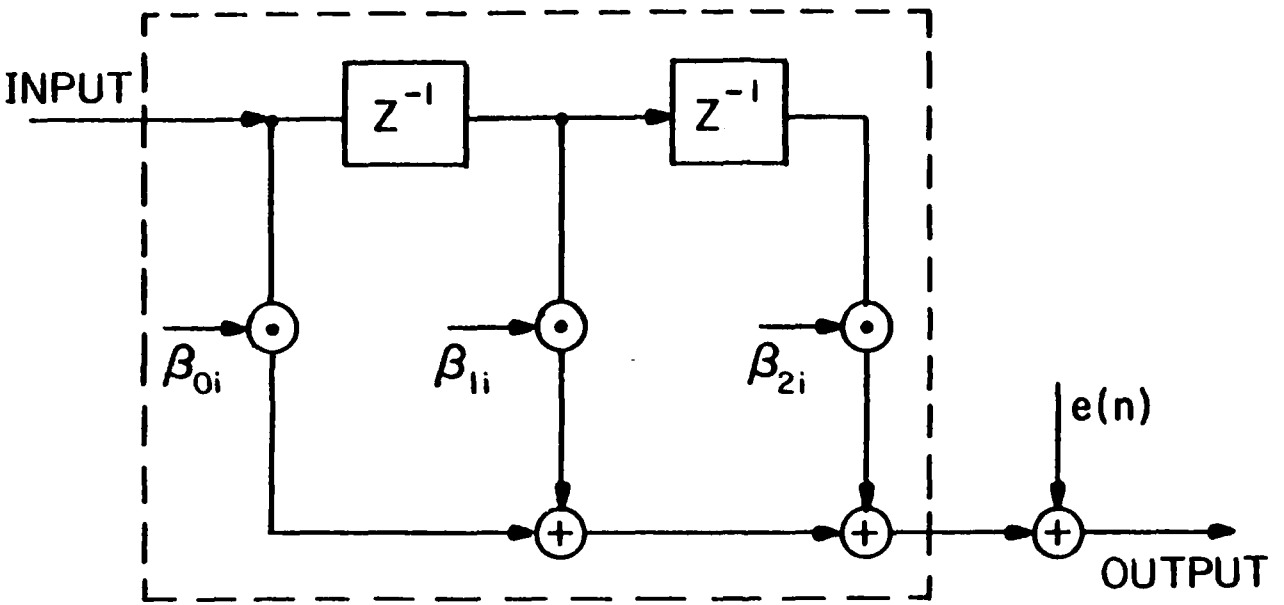


Figure 3-2. Second-order SGK filter noise model

$$t_{j,i}(\ell,m) = \begin{cases} \begin{bmatrix} 0 & c_1 & 0 \\ 0 & c_2 & 0 \\ 0 & c_3 & 0 \end{bmatrix} & \text{for column convolution} \\ \begin{bmatrix} 0 & 0 & 0 \\ r_1 & r_2 & r_3 \\ 0 & 0 & 0 \end{bmatrix} & \text{for row convolution} \end{cases} \tag{3-10}$$

for i=1,2,...,2Q. Figure 3-3 shows the roundoff noise model for the SVD/SGK convolution filter. The mean and variance of the error sequence e(n) can be shown to be

$$\begin{aligned} m_e &= 0 \\ \sigma_e^2 &= \frac{2^{-2b}}{12} \end{aligned} \tag{3-11}$$

We assume that the rounding is performed with (b+1) bits word-length. In this model, a given error seguence e(n) is filtered by succeeding sections, so that the output noise variance will depend upon the ordering of the second order SGK filters.

Let us define $g_{j,i}(\ell,m)$ to be the impulse response from the noise source $e_i(n)$ to the output. Thus,

$$g_{j,i}(\ell,m)=t_{j,i+1}(\ell,m)\circledast\circledast t_{j,i+2}(\ell,m)\circledast\circledast\ldots\circledast\circledast t_{j,2Q}(\ell,m) \tag{3-12}$$

The mean and variance of the roundoff noise are then given

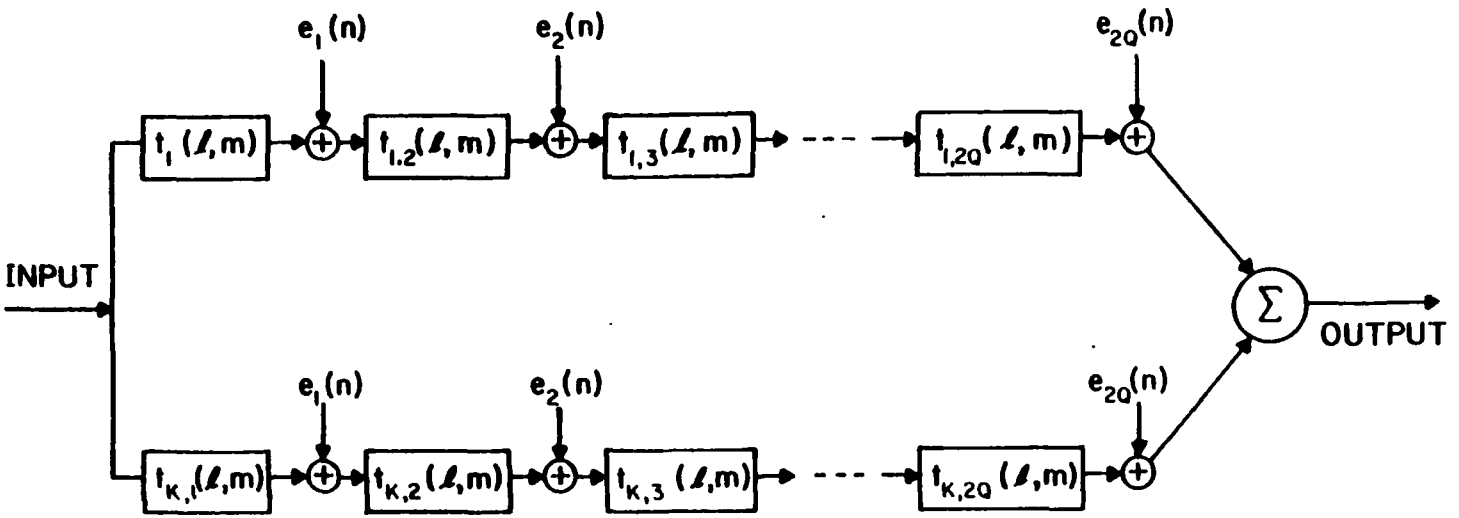


Figure 3-3. Roundoff noise model for SVD/SGK convolution

by

$$m_{e_i} = 0$$

$$\sigma^2_{e_i} = \sigma^2_e \sum_{\ell} \sum_{m} |g_{j,i}(\ell,m)|^2 \tag{3-13}$$

and the total noise variance is the sum of each noise variance of the 3x1 SGK filter. Therefore,

$$\sigma^2_j = \sigma^2_e \sum_{i=1}^{2Q} \left[\sum_{\ell} \sum_{m} |g_{j,i}(\ell,m)|^2 \right] \tag{3-14}$$

If an impulse response $\underline{H}$ is approximated by K singular values, then the total output noise variance due to rounding is

$$\sigma^2_{total} = \sum_{j=1}^{K} \sigma^2_j = \sigma^2_e \sum_{j=1}^{K} \left\{ \sum_{i=1}^{2Q} \left[\sum_{\ell} \sum_{m} |g_{j,j}(\ell,m)|^2 \right] \right\} \tag{3-15}$$

If the two-dimensional impulse response $g_{j,i}(\ell,m)$ consists of N_1 SGK filters for the columns and $(2Q-i-N_1)$ SGK filters for the rows, then $g_{j,i}(\ell,m)$ can be rewritten as

$$g_{j,i}(\ell,m) = g^c_{j,i}(\ell) g^r_{j,i}(m) \tag{3-16}$$

In vector form, Eq. (3-16) is equivalent to

$$\underline{g}_{j,i} = \underline{g}^{c}_{j,i}(\underline{g}^{r}_{j,i})^{T} \tag{3-17}$$

where $\underline{g}^{c}_{j,i}$ and $\underline{g}^{r}_{j,i}$ are one-dimensional impulse responses obtained by convolving N_1 SGK filters for the columns, and $(2Q-i-N_1)$ SGK filters for the rows, respectively. If $\underline{g}_{j,i}$ consists of only SGK filters for the columns or the rows, then $\underline{g}^{c}_{j,i}$ and $\underline{g}^{c}_{j,i}$ should be

$$\underline{g}^{c}_{j,i} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{3-18a}$$

or

$$(\underline{g}^{r}_{j,i})^{T} = [0 \quad 1 \quad 0] \tag{3-18b}$$

Note that

$$\sum_{\ell}\sum_{m}|g_{j,i}(\ell,m)|^2 = \sum_{\ell}|g^{c}_{j,i}(\ell)|^2 \sum_{m}|g^{r}_{j,i}(m)|^2 \tag{3-19}$$

Substituting Eq. (3-19) into Eq. (3-20), we obtain

$$\sigma^2_{total} = \frac{2^{-2b}}{12}\sum_{j=1}^{K}\left\{\sum_{i=1}^{2Q}\left[\sum_{\ell}|g^{c}_{j,i}(\ell)|^2 \sum_{m}|g^{r}_{j,i}(m)|^2\right]\right\} \tag{3-20}$$

Equation (3-20) is a theoretical formula predicting roundoff noise with (b+1) bits word-length. Its validity will be demonstrated in Chapter 5.

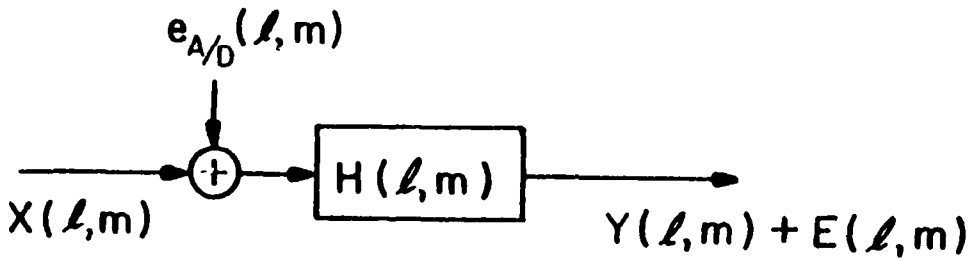
3.3.2 A/D Error

Next an attempt has been made to show that the input A/D noise is negligible compared to the roundoff noise. Again, the statistical model is chosen, and the input quantization is considered as an injection of additive noise to the input. The noise quantitites are uniformly distributed over one quantization interval and statistically independent. Since the first place where quantization of the input signal may take place is at the A/D converter, the A/D noise effect is independent of the structure we used to realize the filter. Figure 3-4 describes the statistical model for A/D noise.

If the quantizer has a word-length of (t+1) bits, then the input to the actual filter is $x(\ell,m)+e_{A/D}(\ell,m)$, where $e_{A/D}(\ell,m)$ is the quantization error, bounded by $-\frac{2^{-t}}{2} \le e_{A/D}(\ell,m) \le \frac{2^{-t}}{2}$. Let us define the output error array, $E(\ell,m)$, as

$$E(\ell,m) = H(\ell,m) \circledast\circledast e_{A/D}(\ell,m) \qquad (3\text{-}21)$$

Since the filter is linear, it can be shown that $E(\ell,m)$ has



$$Y(\ell, m) = X(\ell, m) \circledast\circledast H(\ell, m)$$

$$E(\ell, m) = e_{A/D}(\ell, m) \circledast\circledast H(\ell, m)$$

Figure 3-4. A/D noise model

zero mean and variance given by

$$\sigma^2_{A/D} = \sigma^2_t \sum_{\ell} \sum_{m} |H(\ell,m)|^2 \qquad (3\text{-}22a)$$

where

$$\sigma^2_t = \frac{2^{-2t}}{12} \qquad (3\text{-}22b)$$

It is noted that the filter has been normalized, so that

$$\sum_{\ell} \sum_{m} H(\ell,m) = 1 \qquad (3\text{-}23)$$

Such a normalized filter will not change image contrast between input and output. Therefore, it is obvious that

$$\sum_{\ell} \sum_{m} |H(\ell,m)|^2 \leq \sum_{\ell} \sum_{m} H(\ell,m) = 1 \qquad (3\text{-}24)$$

Using Eqs. (3-17) and (3-19), and assuming that the quantizer has the same word-length as a multiplier, it can be shown that

$$\sigma^2_{A/D} \leq \sigma^2_e \qquad (3\text{-}25)$$

It is shown that the A/D noise is smaller than or equal to that of roundoff noise. In general, A/D noise is negligible compared to roundoff noise.

It is necessary to remark on the effect of filter coefficient quantization. Although zero location and frequency response sensitivities to coefficient changes can readily be obtained, no general statistical analysis of the type given in Section 3-3 has been obtained to describe the cascade form of FIR filters. Herrmann and Schuessler [3-9] worked on this problem only experimentally, not theoretically.

3.4 Conclusion

The accuracy of a digital filter is limited by the finite word-length used in its implementation. When a digital filter is implemented with special-purpose hardware, one is usually interested in determining the minimum word-length needed for a specified performance accuracy. Also, word-length is an important factor in determining the complexity of hardware and speed. Thus, it is very important to understand the effect of quantization.

In this chapter, attempts have been made to analyze relevant effects of using fixed-point arithmetic for SVD/SGK convolution from a statistical viewpoint. Our consideration of finite word-length effect began with a discussion of the various methods of number representation that are commonly used in digital system. The following discussion focused on three common source of errors caused by implementation with finite word-length. Then, we showed

how a statistical analysis can estimate the effects of quantization in SVD/SGK convolution. Statistical methods were shown to be very efficient in systems with non-deterministic signals. It was also shown that roundoff noise is of major concern in digital implementation, and a theoretical formula to predict total roundoff noise variance of SVD/SGK convolution was derived. The A/D noise was shown to be negligible in our case. Finally, the dependence of the roundoff noise on section ordering was demonstrated. The discussion of section ordering and a dynamic range consideration in the fixed-point arithmetic is the subject of the next chapter.

CHAPTER 4

SCALING AND SECTION ORDERING

4.1 Introduction

In the previous chapter, a theoretical formula for predicting roundoff noise variance was derived. One important constraint should be imposed on the implementation with fixed-point arithmetic. There is a finite dynamic range of fixed-point arithmetic. To ensure that the final output be correct, overflow at the output of any second-order SGK filter must be avoided. If the output of each section (SGK filter) exceeds the finite dynamic range of the filter, undesired signal distortion will be introduced to the output. For example, given the dynamic range of (-1.0, 1.0), adding two numbers may result in a number that is not within the given range. Truncating or rounding operations that assign the limit value to the result (say -1.0, or 1.0) introduce an error. This problem directs attention to the need for a scaling procedure for the filter parameters of each SVD/SGK section in order to prevent overflow.

Another issue, section ordering, is also important to

minimize roundoff noise. As seen in Eq. (3-15), the total roundoff noise has a strong dependence on section ordering, i.e., $g_{j,i}(\ell,m)$ will be different if the ordering is different. For example, Schussler [4-1] has demonstrated that a FIR filter with a length of 33, ordered one way, produces $\sigma^2 = 2.40^2$, where Q is the quantization step, while ordered another way yields $\sigma^2 = 1.5\times10^8 Q^2$. Although this experiment demonstrated two extreme cases, it clearly shows the importance of section ordering in cascade form FIR filters. Since the respective difference is so large, determining the minimum roundoff noise ordering is essential.

Unfortunately, attempts to find optimal ordering become impractical since, given M sections, there are M! possible orderings. Even for a moderate value of M, say M = 7, searching 5040 possible orderings is very time-consuming. Furthermore, due to the analytical complexity of Eq. (3-15), no analytical approach to finding an optimal ordering seems possible.

Chan and Rabiner [4-2] investigated the section ordering problem of one-dimensional, cascade form FIR filters quite intensively and reported their results, based on the experiment, as follows:

1) Most orderings have very low noise compared to the maximum possible value. More specific, for a FIR

lowpass filter with length 11, they showed that approximately two-thirds of the orderings have noise variance less than 4% of the maximum, of which nine-tenth have noise variance less than 14% of the maximum.

2) There is a large gap between small and large noise variance distribution, and the noise values within the gap are produced by very few orderings.

Their conclusions are encouraging. Since the large majority of possible orderings are very close to the minimum noise variance ordering, finding a suboptimal ordering is possible with reasonable computations. Instead of finding a time-consuming optimal ordering, it may be far more practical to use a suboptimal ordering method that can rapidly determine an ordering that is close to the optimal. Furthermore, the reduction in roundoff noise gained by finding the optimum ordering is very small, compared to a good suboptimal ordering.

Based upon their experiments, Chan and Rabiner proposed a simple one-dimensional ordering algorithm [4-2], which has proven to be very efficient in minimizing roundoff noise variance.

The purpose of this chapter is to discuss scaling

procedures and ordering algorithms for SVD/SGK convolution. Two existing scaling methods, sum and L_p-norm scaling, are discussed, and applications to SVD/SGK convolution are given in Section 4-2. A brief review of the Chan and Rabiner ordering algorithm and its generalization to SVD/SGK convolution are described in Section 4-3.

4.2 Scaling Procedure

Scaling is important because the computational dynamic range sets a practical limit to the maximum value of signal levels representable in the filter. The theoretical basis for the scaling procedure chosen here is Jackson's work [4-3], commonly referred to as sum scaling. To formulate the required overflow constraints, let us assume that an input signal x(n,m) is bounded in magnitude by 1.0.

We shall consider a scaling procedure in which a (2Q+1)x(2Q+1) FIR filter is implemented by SVD/SGK convolution. There are 2Q 3x1 SGK filters for the columns and rows in each SVD expansion stage. To simplify notation, only one SVD expansion stage will be considered. Therefore, the subscript j will be dropped.

We will define $f_i(\ell,m)$ to be the impulse response from the input to the i-th section. The z-transform of $f_i(\ell,m)$ can be written as

$$F_i(z_1,z_2) = \sum_{\ell}\sum_{m} f_i(\ell,m) z_1^{-\ell} z_2^{-m}$$
$$= \prod_{p=1}^{i} T_p(z_1,z_2) \qquad (4\text{-}1)$$

where $T_p(z_1,z_2)$ is the z-transform of $t_p(\ell,m)$, as defined in Eq. (3-10). Let S_i be the scaling factor for the i-th section and $T_i'(z_1,z_2)$ be a scaled z-transform of $T_i(z_1,z_2)$. Then

$$T_i'(z_1,z_2) = S_i T_i(z_1,z_2) \qquad (4\text{-}2)$$

and the scaled transfer function from the input to the i-th section is

$$F_i'(z_1,z_2) = \sum_{\ell}\sum_{m} f_i'(\ell,m) z_1^{-\ell} z_2^{-m} \qquad (4\text{-}3a)$$

or

$$F_i'(z_1,z_2) = \prod_{p=1}^{i} T_p'(z_1,z_2) = \prod_{p=1}^{i} S_i \prod_{p=1}^{i} T_p(z_1,z_2) \qquad (4\text{-}3b)$$

Letting $v_i(\ell,m)$ be the output at the i-th section, $v_i(\ell,m)$ is obtained by convolving the input array $x(\ell,m)$ with the impulse response $f_i'(\ell,m)$. Thus

$$v_i(\ell,m) = \sum_{p}^{\ell}\sum_{q}^{m} x(p,q) f_i'(\ell-p+1, m-q+1) \qquad (4\text{-}4)$$

and $|v_i(\ell,m)|$ is bounded by

$$|v_i(\ell,m)| \le |x(\ell,m)|_{max} \sum_p \sum_q |f_i'(\ell,m)| \tag{4-5}$$

$$= |x(\ell,m)|_{max} \prod_{p=1}^{i} |S_p| \sum_p \sum_q |f_i(p,q)|$$

Therefore, a necessary and sufficient condition on the scale factor to ensure that the output of each section is bounded in magnitude by 1.0 is that

$$|x(\ell,m)|_{max} \prod_{p=1}^{i} |S_p| \sum_p \sum_q |f_i(p,q)| \le 1.0 \tag{4-6}$$

Since $|x(\ell,m)|$ is bounded by 1.0, Eq. (4-6) is equivalent to

$$\prod_{p=1}^{i} |S_p| \le \left[\sum_p \sum_q |f_i(p,q)| \right]^{-1} \tag{4-7}$$

The scaling procedure of Eq. (4-7), satisfied with equality, is called sum scaling.

Another scaling procedure, referred to as L_p-norm scaling, was also introduced by Jackson [4-3]. Note that the i-th section output $v_i(\ell,m)$ satisfies the condition

$$v_i(\ell,m) = \frac{1}{4\pi^2} \int_{-\pi}^{\pi} \int_{-\pi}^{\pi} \mathfrak{I}_i'(u,v) \mathfrak{X}(u,v) du dv \tag{4-8}$$

where $\mathcal{F}'(u,v)$ and $\mathcal{X}(u,v)$ are the Fourier transform of $f_i'(\ell,m)$ and $x(\ell,m)$ respectively. Here we assume that the input $x(\ell,m)$ is a deterministic signal.

The L_p-norm of a Fourier transform $\mathcal{X}(u,v)$ is defined as

$$\| H\|_p = \left[\frac{1}{4\pi^2} \int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |\mathcal{X}(u,v)|^p dudv \right]^{1/p} \qquad (4\text{-}9)$$

Equations (4-8) and (4-9) immediately lead to the relation

$$|v_i(\ell,m)| \le \frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |\mathcal{F}_i'(u,v)\mathcal{X}(u,v)|dudv = \| F_i X\|_1 \qquad (4\text{-}10)$$

Applying the Schwartz inequality to Eq.(4-10) yields the relation

$$|v_i(\ell,m)|^2 \le \frac{1}{4\pi^2}\int_{-\pi}^{\pi} |\mathcal{F}_i'(u,v)|^2 dudv \cdot \int_{-\pi}^{\pi} |\mathcal{X}(u,v)|^2 dudv \qquad (4\text{-}11a)$$

or

$$|v_i(\ell,m)|^2 \le \| F_i'\|_2 \, \| X\|_2 \qquad (4\text{-}11b)$$

In general, it can be shown that

$$\| F_i' \cdot X \| \le \| F_i'\|_p \cdot \| X\|_q \qquad (4\text{-}12)$$

for

$$\frac{1}{p} + \frac{1}{q} = 1$$

and $p,q \geq 1$. Therefore, with L_p-norm scaling, the required condition and preventing overflows is satisfied by

$$|v_i(\ell,m)| \leq \|F_i'\|_p \; \|X\|_q \tag{4-13}$$

Based on Eq. (4-13), a sufficient condition for scaling can be given if one has knowledge of the L_p-norm of the input signal. One particularly interesting case is $p = \infty$ and $q = 1$. In this case, the input signal should be bounded by

$$\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi}|\chi(u,v)|\,dudv \leq 1.0 \tag{4-14}$$

Then, the necessary and sufficient condition on the scale factor to guarantee that the output of each section is bounded in magnitude by 1.0 is that

$$\|F_i'\|_\infty = \left[\underset{\substack{-\pi\leq u\leq\pi \\ -\pi\leq v\leq\pi}}{\text{MAX}} \; |\mathcal{F}_i'(u,v)|\right] \leq 1.0 \tag{4-15}$$

which is equivalent to

$$\prod_{j=1}^{i} S_j \leq \left[\underset{\substack{-\pi\leq u\leq\pi \\ -\pi\leq v\leq\pi}}{\text{MAX}} \; |\mathcal{F}_i'(u,v)|\right]^{-1} \tag{4-16}$$

The scaling procedure of Eq. (4-16), satisfied with equality, is known as peak scaling [4-3].

The two scaling procedures discussed above are summarized as follows. If the input signal is bounded by $|x(\ell,m)| \leq 1.0$, then each scale factor S_i can be computed according to the following procedure:

1. Compute

$$\sigma_i^2 = \sum_{\ell}\sum_{m} |f_i(\ell,m)| \quad \text{for } i=1,2,\ldots,2Q \tag{4-17a}$$

2. Then

$$S_i = \begin{cases} \dfrac{1}{\sigma_i} & \text{for } i=1 \\ \dfrac{\sigma_{i-1}}{\sigma_i} & \text{for } i=2,3,\ldots,2Q \end{cases} \tag{4-17b}$$

If input signal is bounded such that

$$\frac{1}{4\pi^2}\int_{-\pi}^{\pi}\int_{-\pi}^{\pi} |\chi(u,v)|\,du\,dv \leq 1.0 \tag{4-18a}$$

then each scale factor S_i is computed as follows:

1. Compute

$$\gamma_i = \left[\underset{\substack{-\pi \le u \le \pi \\ -\pi \le v \le \pi}}{\text{MAX}} |\mathfrak{F}_i(u,v)| \right] \quad \text{for } i=1,2,\ldots,2Q \tag{4-18b}$$

2. Then

$$S_i = \begin{cases} \dfrac{1}{\gamma_i} & \text{for } i=1 \\ \dfrac{\gamma_{i-1}}{\gamma_i} & \text{for } i=2,3,\ldots,2Q \end{cases} \tag{4-18c}$$

But, it should be noted here that Eq. (4-16) cannot be used in the case of a random input signal, because if the signal is random, its Fourier transform does not exist. Instead of using the Fourier transform, the equivalent condition can be obtained with the appropriate power spectral density and autocorrelation function [4-3].

Experimental results indicate that the two scaling procedures yield noise variances that are similar [4-2]. In general, sum scaling is much simpler to perform than peak scaling in FIR filter cases. With peak scaling, one must find the maxima of the $|\mathfrak{F}_i(u,v)|$ for all i. Even using the FFT algorithm will require more computations than finding $\sum_{\ell}\sum_{m}|f_i(\ell,m)|$ for all i. It has been claimed that sum scaling is too conservative to be used in IIR filter cases [4-4]. But this is not of major concern in the case of a FIR filter. Therefore, in order to save computation time, we shall focus on sum scaling throughout

this study.

The sum scaling of Eq. (4-7) requires computation of the two-dimensional impulse response $f_i(\ell,m)$ for all i. Since each SVD-expanded matrix $\underline{H}_j$ of Eq. (2-17) is separable, further simplification is possible for SVD/SGK convolution. Note that each separable matrix $\underline{H}_j$ is an outer product of one-dimensional column and row convolution operators $\underline{c}_j$ and $\underline{r}_j$. Instead of applying the sum scaling by computing $f_i(\ell,m)$, the same result will be obtained by applying the sum scaling to $\underline{c}_j$ and $\underline{r}_j$ independently. The following Lemma will generalize the above argument.

Lemma: If a two-dimensional separable impulse response matrix $\underline{H}$ is realized in the one-dimensional cascade form, sum scaling can be applied to the one-dimensional convolution operators $\underline{c}$ and $\underline{r}$ independently.

Proof: In the SVD/SGK convolution system, there are 0 3 x 1 SGK filters for the columns and rows of the input image. Given a certain ordering, there are N_1 3 x 1 SGK filters for the columns and $(i-N_1)$ 3 x 1 SGK filters for the rows from input to the i-th section. Let us assume the i-th section is a filter for the column. From the sum scaling of Eq. (4-17), we have

$$S_i = \frac{\sigma_{i-1}}{\sigma_i} = \frac{\sum_{\ell}\sum_{m}|f_{i-1}(\ell,m)|}{\sum_{\ell}\sum_{m}|f_i(\ell,m)|} = \frac{\sum_{\ell}|f^c_{i-1}(\ell)|\sum_{m}|f^r_{i-1}(m)|}{\sum_{\ell}|f^c_i(\ell)|\sum_{m}|f^r_i(m)|} \quad (4\text{-}19)$$

where $f^c_i(\ell)$ and $f^r_i(m)$ are one-dimensional impulse responses obtained by convolving the N_1 filters of the columns and $(i-N_1)$ filters of the rows. The superscripts c,r are associated with column and row, respectively. But,

$$\sum_m |f^r_{i-1}(m)| = \sum_m |f^r_i(m)| \quad (4\text{-}20)$$

Therefore,

$$S_i = \frac{\sum_{\ell}|f^c_{i-1}(\ell)|}{\sum_{\ell}|f^c_i(\ell)|} \quad (4\text{-}21)$$

which is equivalent to one-dimensional sum scaling.

By the Lemma, two-dimensional sum scaling is shown to be equivalent to two one-dimensional sum scaling operations. The same Lemma can be applied to peak scaling. Since $f_i(\ell,m)$ is separable, then

$$\mathcal{F}_i(u,v) = \mathcal{F}^c_i(u)\mathcal{F}^r_i(v) \quad (4\text{-}22)$$

Therefore,

$$\left[\max_{\substack{-\pi\le u\le\pi\\ -\pi\le v\le\pi}} |\mathcal{F}_i(u,v)|\right]=\left[\max_{-\pi\le u\le\pi} |\mathcal{H}_i^c(u)|\right]\left[\max_{-\pi\le v\le\pi} |\mathcal{H}_c^r(v)|\right] \qquad (4\text{-}23)$$

4.3 Section Ordering

The next step, given the scaling procedure chosen, is to choose an ordering for the sections to minimize the total roundoff noise. As an approach to determine an ordering algorithm for SVD/SGK convolution, a one-dimensional ordering algorithm for the cascade form FIR filter will be introduced. If a FIR filter of size (2Q+1) is realized in cascade form, there are Q sections of second-order filters and Q! possible orderings. If we define $b_i(k)$ for i=1,2,...,Q to be the impulse response from the (i+1)-st section to the output, the total roundoff noise variance can be shown to be [4-2]

$$\sigma_r^2 = \frac{2^{-2b}}{12} \sum_{i=1}^{Q} \left[\sum_k |b_i(k)|^2\right] \qquad (4\text{-}24)$$

Here we assume that the rounding is performed only after the products are represented in full accuracy. The best ordering will minimize the total output noise variance. Based on the Chan and Rabiner experiment, the proposed algorithm is summarized as follows [4-2]:

Beginning with i = Q, assign the i-th section, together with the section already assigned, that yields the smallest possible value for $\sum_k |b_{i-1}(k)|^2$.

This algorithm is suboptimal since the algorithm minimizes the output noise variance from individual sections instead of minimizing the sum of the output noise variance. However, in all cases tested, the algorithm has proved to yield section ordering very close to the optimum ordering because a large majority of possible orderings yield small output noise variance, as discussed before.

In SVD/SGK convolution, there are a total of 2Q 3 x 1 SGK filters. Searching (2Q)! possible ordering is an enormous task. But we shall see, based on the existing theory, the generalization of a one-dimensional ordering algorithm for SVD/SGK convolution is possible, and the proposed algorithm will prove to be efficient and simple.

Let us rewrite the output noise variance formula for SVD/SGK convolution, as derived in Eqs. (3-15) and (3-20), as

$$\sigma^2_{total} = \frac{2^{-2b}}{12} \sum_{j=1}^{K} \left\{ \sum_{i=1}^{2Q} \left(\sum_{\ell} \sum_{m} |g_{j,i}(\ell,m)|^2 \right) \right\} \qquad (4\text{-}25)$$

$$\sigma^2_{total} = \frac{2^{-2b}}{12} \sum_{j=1}^{K} \left\{ \sum_{i=1}^{2Q} \left(\sum_{\ell} |g^c_{j,i}(\ell)|^2 \sum_{m} |g^r_{j,i}(m)|^2 \right) \right\} \qquad (4\text{-}26)$$

where $g_{j,i}(\ell,m)$, $g^{c}_{j,i}(\ell)$, and $g^{r}_{j,i}(m)$ are already defined in Section 3-3. Again, only one SVD expansion term will be considered; therefore, the subscript j will be dropped.

Using Eq. (4-25), it is quite simple to extend the Chan and Rabiner ordering algorithm to SVD/SGK convolution. But the significance of using Eq. (4-26) to search for an ordering algorithm for SVD/SGK convolution is that the ordering problem can be treated as solving two one-dimensional ordering problems. Since $\sum_{\ell} |g^{c}_{i}(\ell)|^2$ and $\sum_{m} |g^{r}_{i}(m)|^2$ are positive numbers, the following is satisfied:

$$\min\left[\sum_{\ell}|g^{c}_{i}(\ell)|^2\sum_{m}|g^{r}_{i}(m)|^2\right] = \min\left[\sum_{\ell}|g^{c}_{i}(\ell)|^2\right]\min\left[\sum_{m}|g^{r}_{i}(m)|^2\right] \tag{4-27}$$

Rather than minimizing $\sum_{\ell}\sum_{m}|g_{i}(\ell,m)|^2$, an equivalent condition can be obtained by minimizing $\sum_{\ell}|g^{c}_{i}(\ell)|^2$ and $\sum_{\ell}|g^{r}_{i}(m)|^2$ separately. Thus, minimizng $\sum_{\ell}|g^{c}_{i}(\ell)|^2$ and $\sum_{m}|g^{r}_{i}(m)|^2$ is equivalent to two one-dimensional ordering problems. After ordering column and row operators independently, the remaining step is to decide whether the SGK filter on the column or on the row should be assigned at the i-th section.

To show the rationale for the algorithm mathematically, assume that (2Q-i) SGK column operators and

row operators have been already assigned with N_1 column operators and $(2Q-N_1-i)$ row operators. Now, we want to select the i-th section of the SVD/SGK convolution. To simplify the discussion, let us define

$$\alpha_i = \sum_{\ell} |g^c_{i+1}(\ell)|^2 \tag{4-28}$$

$$\beta_i = \sum_{m} |g^r_{i+1}(m)|^2 \tag{4-29}$$

If we had assigned the next filter on the columns to the i-th section, then the output noise variance would be proportional to

$$E^c_i = \beta_i \sum_{\ell} |g^c_i(\ell)|^2 \tag{4-30}$$

If we had assigned the next filter on the rows to the i-th section, the resulting output noise variance would be proportional to

$$E^r_i = \alpha_i \sum_{m} |g^r_i(m)|^2 \tag{4-31}$$

By comparing E^c_i and E^r_i of Eqs. (4-30) and (4-31), one can easily decide whether the filter on the columns or the filter on the rows should be assigned to the i-th section.

Since α_i, β_i for $i=1,2,\ldots Q$ can be obtained as a result of one-dimensional orderings for the column and row

operators, this procedure is far simpler than using Eq. (4-25).

In brief, the proposed ordering algorithm is summarized as follows:

1. Find a one-dimensional ordering to the column and the row operators by using the Chan and Rabiner algorithm and store α_i, β_i for i=1,2,...Q.

2. Beginning with i=2Q, compare E_i^c and E_i^r given by Eqs. (4-30) and (4-31), respectively, to decide whether the filter on the rows or the filter on the columns should be assigned to the i-th section.

This proposed algorithm is also suboptimal in minimizing $\sum_{\ell}\sum_{m}|g_i(\ell,m)|^2$ rather than minimizing $\sum_i \sigma_i$, where

$$\sigma_i = \sum_{\ell}\sum_{m}|g_i(\ell,m)|^2.$$

4.4 Conclusion

In addition to the effect of finite word-length discussed in Chapter 3, the problems of overflow and section ordering to minimize the total roundoff noise are of great importance when a digital filter is realized in cascade form. To prevent overflow, the filter parameters and input signals must be scaled so that no overflow occurs following addition. Proper ordering must also be found for

a filter in cascade form because the output roundoff noise has a strong dependence on the way it is ordered.

Following the discussion of two different scaling methods, sum and L_p-norm scalings, sum scaling was chosen because sum scaling is simple and easy to employ. A detailed sum scaling procedure for SVD/SGK convolution was presented. Because separable matrices result from the SVD expansion of a nonseparable impulse response matrix, the two-dimensional scaling problem turned out to be two one-dimensional scaling problems. The proof was given in a Lemma.

Next, the section ordering problem was considered. Extending the existing one-dimensional suboptimal ordering algorithm proposed by Chan and Rabiner [4-4], a generalized two-dimensional suboptimal ordering algorithm for SVD/SGK convolution was proposed. Because it is actually equivalent to two one-dimensional ordering problems, the proposed ordering algorithm is very simple and fast to compute. The experimental results based on the proposed ordering algorithm, which is shown to be very efficient, will be discussed in Chapter 5.

CHAPTER 5

EXPERIMENTAL RESULTS OF SVD/SGK CONVOLUTION USING FIXED-POINT ARITHMETIC

5.1 Introduction

In this chapter, computer simulation experimental results for SVD/SGK convolution are presented. Two prototype filters with linear phase have been chosen for the experiments. One is a lowpass filter, the other, a bandpass filter. The sizes of the filters are 15x15 and 11x11, respectively. Perspective views of the frequency response of the prototype filters are given in figures 5-1 and 5-2, respectively. Figure 5-3 shows the SVD approximation errors for the expansion of the prototype filters. It is observed that the NMSE decreases very rapidly in both cases. In the case of the lowpass filter, the SVD approximation error with 3-stage expansion is 0.5336 %. In the case of the bandpass filter, the SVD approximation error for a 4-stage expansion is 0.7825 %. Numerical and photographical results related to the outputs of this SVD/SGK convolution when the inputs are random numbers and real image are presented in this chapter.

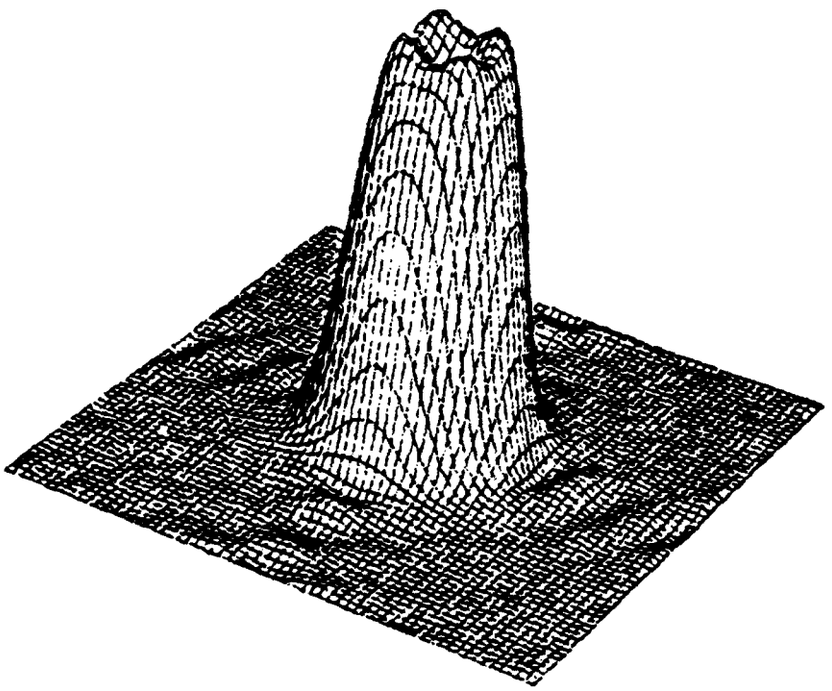


Figure 5-1. Perspective view of the frequency response of the prototype lowpass filter

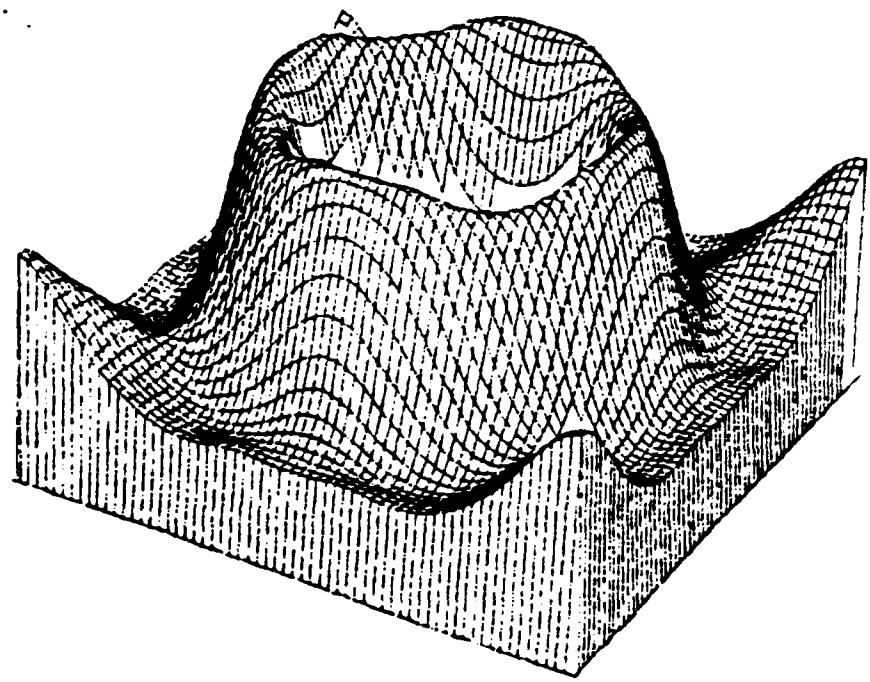


Figure 5-2. Perspective view of the frequency response of the prototype bandpass filter

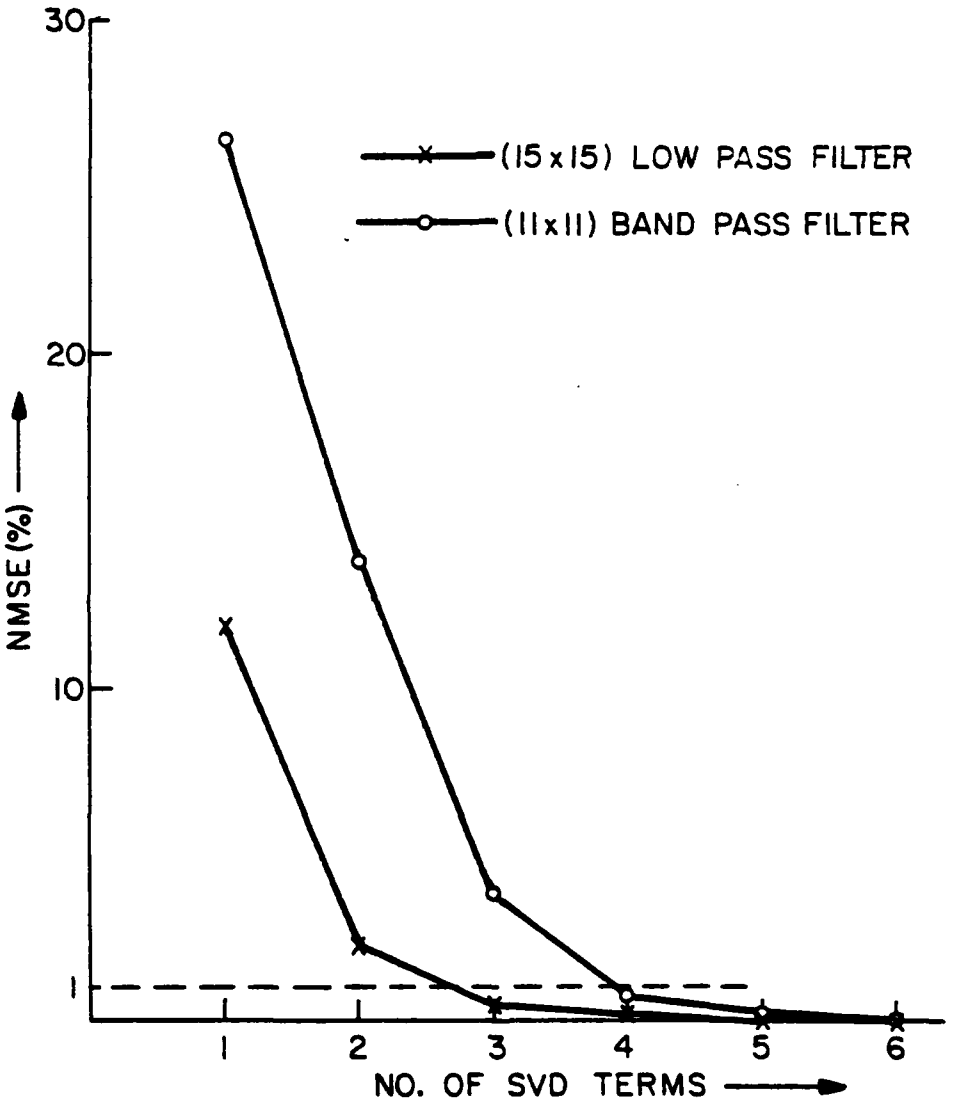


Figure 5-3. NMSE versus number of singular value

5.2 Fixed-Point Arithmetic Experimental Results

Our experiments were made in the following framework. We shall use M to denote the word-length for representing filter coefficients and N to denote the word-length for storing intermediate results. Furthermore, we shall adopt the policy that all signal levels that are representable by given word-length be constrained within the range of (-1,1). A multiplier with input signal level greater than unity may need to be followed by extra accumulators and extra wide adders. Hence, more hardware is required. The number of rounding operations within 3x1 SGK filters is assumed to be one. In other words, since the typical operation performed in convolution is a sum of products, we assume here that the rounding operation is performed only after the products have been summed with full precision. In addition to this, the cascade form of the SVD/SGK convolution requires a proper section ordering. The suboptimal ordering algorithm discussed in Chapter 4 was adopted to minimize roundoff noise. Because of the quadrilateral symmetry of the prototype filters used, the one-dimensional column and row convolution operators obtained from the SVD expansion of $\underline{H}$ were identical. Thus, their cascade forms were identical. The ordering algorithm ended with a perfect interlace scheme; each filter for the rows convolution was followed by a filter for the columns convolution and vice versa. But it can be proved that this

result, although often true, cannot be generalized to all cases of quadrilateral symmetrical filters.

Then, after the ordering procedure, sum scaling was applied to the filter coefficients so that overflow will not occur within filters. Unfortunately, large differences in magnitude among the coefficients causes the scaled filter coefficients to exceed the given dynamic range of word-length. In this case, the filter coefficients were further divided by their maximum coefficient to insure that the scaled filter coefficients lie within the given dynamic range of word-length for the filter coefficents.

5.2.1 Roundoff Error

To confirm the validity of the noise formula of Eq. (3-15), derived in Chapter 3, a uniform density random number array of 46x46 pixels has been generated as an input. The statistical approach used to analyze roundoff noise in Chapter 3 is not practical if the input is deterministic. For this analysis, an image array has been modeled as a Markov process with an adjacent pixel correlation coefficient along lines of 0.95 [5-1]. Furthermore, it has been assumed that the maximum signal magnitude of the input array is unity, so that all signals are represented by given dynamic ranges of word-length.

If the filter size is 15x15, then the output size is

60x60. Because the noise formula is valid only under "steady state" conditions, the actual output is taken over a 32x32 portion of the output array, ignoring a band of width of 14 along each of the four edges of the real output array. The designed SVD/SGK convolution filter was convolved with the given input array in fixed-point arithmetic. The filter coefficients were represented by floating-point with 36 bits of word-length. The standard deviation of the actual errors produced at the output with rounding to N bits was measured and compared with its theoretical estimates computed from the noise formula of Eq. (3-15). The system of Fig. 5-4 was used to measure the value of for various word-length of storage [5-2]. The system $H_{\infty}(z_1, z_2)$ was implemented with floating-point arithmetic with 36 bits of word-length. Table 5-1 shows the experimental results. There is good agreement between the predicted and measured values. This confirms the validity of our model and a statistical approach to analyze the roundoff noise.

5.2.2 Filter Coefficient Quantization Effect

In Chapter 3, the quantization effect of the filter coefficients was shown to be not as severe as that of rounding. Before we present the experimental results, it would be beneficial to discuss the error measurement of a pair of images. A true comparison between a pair of images

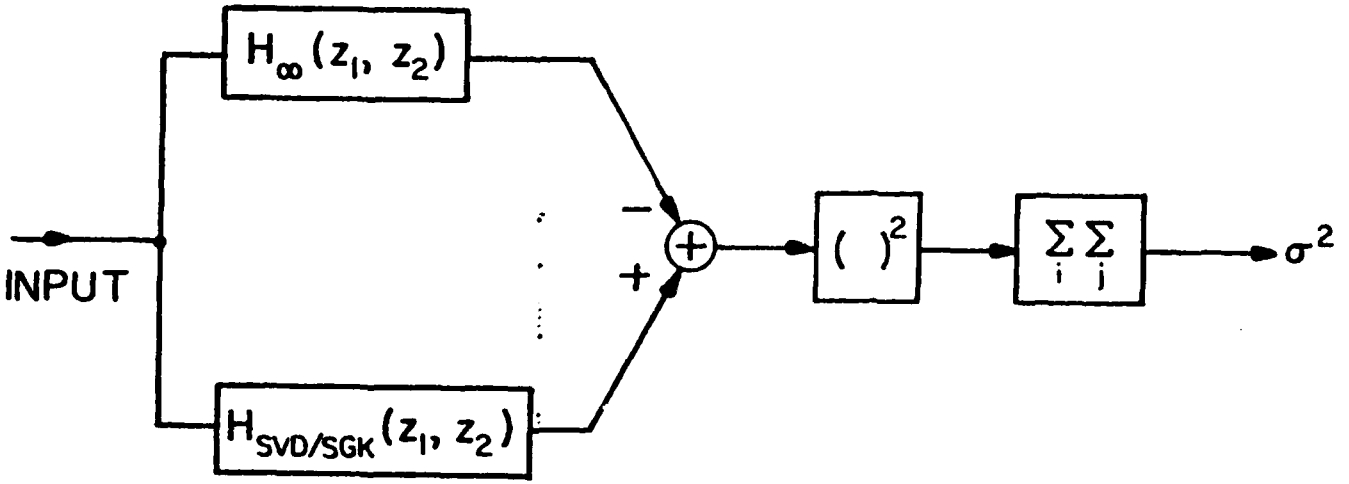


Figure 5-4. Technique measuring variance of the error caused by rounding operations

TABLE 5-1

Standard deviation of output noise caused by rounding operations for a prototype filter

N	Theory	Experiment
8	0.552×10^{-2}	0.719×10^{-2}
10	0.138×10^{-2}	0.191×10^{-2}
12	0.345×10^{-3}	0.465×10^{-3}
14	0.862×10^{-4}	0.117×10^{-4}
16	0.216×10^{-4}	0.291×10^{-4}

Lowpass Filter

N	Theory	Experiment
8	0.329×10^{-1}	0.439×10^{-1}
10	0.823×10^{-2}	0.111×10^{-1}
12	0.206×10^{-2}	0.270×10^{-2}
14	0.514×10^{-3}	0.685×10^{-3}
16	0.129×10^{-3}	0.173×10^{-3}

Bandpass Filter

should follow some objective criteria. It is desirable that the objective criteria be mathematically tractible and reasonably calculable so that they can be used as objective performance functions to evaluate an image processing system. Considerable attention has been paid to the development of such criteria [5-3]. Unfortunately, because of the complexity of the human visual system, there are no universally accepted criteria to measure image fidelity. But, the most commonly used quantitative measure of a pair of image is the normalized mean square error (NMSE), as defined in Chapter 2 [5-4]. We shall use the NMSE as our objective criterion throughout this dissertation. Table 5-2 shows the computed NMSE between floating-point arithmetic with 36 bits of word-length and fixed-point arithmetic with different N and M bits of word-length. In all cases, the results obtained with M =16 bits are close to those with full precision. It is concluded that 16 bits of word-length to quantize filter coefficients is sufficient without reducing filter performance significantly. In Chapter 3, it was shown that the storage required for the filter coefficient is far less than that required for the data. We will then consider that it is more practical to reduce the word-length required for the data storage.

TABLE 5-2

Fixed-point implementation error for various word-length

N \ M	8	12	16	30
8	3.4221	1.4033	1.3197	1.3450
10	3.3241	0.7768	0.3174	0.3082
12	3.3706	0.6223	0.0843	0.0741
14	3.3700	0.5995	0.0221	0.0194
16	3.3693	0.5993	0.0088	0.0048
Floating	3.3690	0.5990	0.00765	0.0000065

Lowpass Filter

N \ M	8	12	16	30
8	7.9396	7.3349	7.1219	7.1115
10	3.9017	1.8639	1.8007	1.7972
12	3.5280	0.6642	0.4529	0.4463
14	3.5184	0.4887	0.1123	0.1081
16	3.5196	0.4842	0.028	0.028
Floating	3.5190	0.4841	0.00435	0.0000191

Bandpass Filter

5.2.3 Output Image Comparison

In order to evaluate the performance of the SVD/SGK convolution more precisely, let us define the following NMSE factors. Assuming that $\underline{G}$ and $\underline{F}$ are output and input arrays, respectively, then we shall use the following notation

$$\underline{G} = \underline{F} \circledast\circledast \underline{H} \tag{5-1a}$$

$$\hat{\underline{G}}_{SVD} = \underline{F} \circledast\circledast \hat{\underline{H}}_{SVD} \tag{5-1b}$$

$$\hat{\hat{\underline{G}}}_{SVD/SGK} = \underline{F} \circledast\circledast \hat{\hat{\underline{H}}}_{SVD/SGK} \tag{5-1c}$$

where $\underline{H}$ is a prototype impulse response, $\hat{\underline{H}}_{SVD}$ is the approximation of $\underline{H}$ by retaining a few dominant terms in the SVD expansion of $\underline{H}$, and $\hat{\hat{\underline{H}}}_{SVD/SGK}$ is the SVD/SGK convolution realization of $\hat{\underline{H}}_{SVD}$. In Eq. (5-1c), the term on the right has been computed using fixed-point arithmetic. Then we define

$$\varepsilon_1 = \left[\frac{\sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2}{\sum_i \sum_j |G(i,j))|^2} \right]^{1/2} \tag{5-2a}$$

$$\varepsilon_2 = \left[\frac{\sum_i \sum_j |\hat{G}_{SVD}(i,j) - \hat{\hat{G}}_{SVD/SGK}(i,j)|^2}{\sum_i \sum_j |\hat{G}_{SVD}(i,j)|^2} \right]^{1/2} \quad (5\text{-}2b)$$

(5-2b)

$$\varepsilon_3 = \left[\frac{\sum_i \sum_j |G(i,j) - \hat{\hat{G}}_{SVD/SGK}(i,j)|^2}{\sum_i \sum_j |G(i,j)|^2} \right]^{1/2} \quad (5\text{-}2c)$$

Two errors are involved in SVD/SGK convolution with fixed-point arithmetic: ε_1 is the error caused by the SVD approximation, and ε_2 is the error due to fixed-point arithmetic. ε_3 is the total error. Table 5-3 summarizes the computed NMSE with different word-length of data storage. In this experiment, the filter coefficients were quantized with 16 bits, and the input was a random array with correlation coefficients of 0.95. Returning to Eq. (5-2), we shall derive an upper bound of the total error ε_3. Since the total error ε_3 is contributed by ε_1 as well as ε_2, this bound will be very useful in SVD/SGK convolution implementation. Let us rewrite Eq. (5-2) in terms of a matrix Euclidean norm, which is defined to be

$$\| \underline{G} \| = \left[\sum_i \sum_j |G(i,j)|^2 \right]^{1/2} \quad (5\text{-}3)$$

Hence, Eq. (5-2) can be rewritten as

$$\varepsilon_1^2 \| \underline{G} \|^2 = \| \underline{G} - \hat{\underline{G}}_{SVD} \|^2 \quad (5\text{-}4a)$$

$$\varepsilon_2^2 \| \hat{\underline{G}}_{SVD} \|^2 = \| \hat{\underline{G}}_{SVD} - \hat{\hat{\underline{G}}}_{SVD/SGK} \|^2 \quad (5\text{-}4b)$$

$$\varepsilon_3^2 \, \| \underline{G} \|^2 = \| \underline{G} - \hat{\hat{\underline{G}}}_{SVD/SGK} \|^2 \qquad (5\text{-}4c)$$

But note that

$$\| \underline{G} - \hat{\hat{\underline{G}}}_{SVD/SGK} \|^2 = \| \underline{G} - \hat{\underline{G}}_{SVD} + \hat{\underline{G}}_{SVD} - \hat{\hat{\underline{G}}}_{SVD/SGK} \|^2 \qquad (5\text{-}5)$$

Using the Schwartz inequality on the right hand side of Eq. (5-5), we have

$$\| \underline{G} - \hat{\hat{\underline{G}}}_{SVD/SGK} \|^2 \leq (\| \underline{G} - \hat{\underline{G}}_{SVD} \| + \| \hat{\underline{G}}_{SVD} - \hat{\hat{\underline{G}}}_{SVD/SGK} \|)^2 \qquad (5\text{-}6)$$

Substituting Eq. (5-4) into Eq. (5-6) results in

$$\varepsilon_3^2 \| \underline{G} \|^2 \leq \varepsilon_1 \| \underline{G} \|^2 + 2\varepsilon_1\varepsilon_2 \| \underline{G} \| \| \hat{\underline{G}}_{SVD} \| + \varepsilon_2^2 \| \hat{\underline{G}}_{SVD} \|^2 \qquad (5\text{-}7a)$$

Therefore

$$\varepsilon_3 \leq \varepsilon_1 + \varepsilon_2 \frac{\| \hat{\underline{G}}_{SVD} \|}{\| \underline{G} \|} \simeq \varepsilon_1 + \varepsilon_2 \qquad (5\text{-}7b)$$

since

$$\| \hat{\underline{G}}_{SVD} \| \simeq \| \underline{G} \| \qquad (5\text{-}7c)$$

Returning to Table 5-3, we can see that the ε_3 error never exceeds the bound given by Eq. (5-7b). However, the fixed-point implementation error ε_2 could be reduced to less than 1.0 % NMSE with 12 bits word-length of storage. The ε_1 error is dominant in the bandpass filter case.

TABLE 5-3

Summary of experiment

ε_i \ N	8	10	12	14	16	∞
ε_1	0.0243	0.0243	0.0243	0.0243	0.0243	0.0243
ε_2	1.3451	0.3174	0.0843	0.0195	0.0048	0.0000065
ε_3	1.3445	0.3174	0.0883	0.0297	0.0249	0.0243

Lowpass Filter

ε_i \ N	8	10	12	14	16	∞
ε_1	3.4981	3.4981	3.4981	3.4981	3.4981	3.4981
ε_2	7.1115	1.7972	0.4464	0.1082	0.0284	0.0000191
ε_3	7.9674	4.0326	3.5160	3.4989	3.4984	3.4981

Bandpass Filter

Obviously, the ε_1 error decreases as more terms are retained in the SVD expansion. Furthermore, there is no reason to believe that ε_1 will be the same order as ε_k of Eq. (2-19). For instance, ε_k of the bandpass filter is 0.7825 % with a 4-stage expansion, But, ε_1 is 3.498 %. But, the situation is quite opposite in the lowpass filter cases. Appendix A describes the relation between the ε_k and ε_1 errors. As shown in Eq. (A-16) of Appendix A, the ε_1 error is mainly attributed to the mean difference between $\underline{G}$ and $\hat{\underline{G}}_{SVD}$. Given fixed ε_k, the ε_1 error increases as the mean difference increases.

The prime goal of this error analysis is to reduce the error and to force the SVD/SGK processed output closer to the direct processed output. If we correct the output so that $m_{\hat{g}_{SVD}}$ equals to m_g, then the ε_1 error is the same order as ε_k. In the following, we shall develop a simple algorithm to force the mean difference equal to zero.

Assuming the filter is time-invariant and linear and m_f is the mean of the input array, then

$$m_g = m_f \sum_i \sum_j H(i,j) \tag{5-8}$$

But, the prototype impulse response matrix $\underline{H}$ is normalized such that $\sum_i \sum_j H(i,j) = 1$, therefore,

$$m_g = m_f \tag{5-9}$$

Also,

$$m_{\hat{g}_{SVD}} = m_f \sum_i \sum_j \hat{H}_{SVD}(i,j) \tag{5-10a}$$

where

$$\underline{\hat{H}}_{SVD} = \sum_{i=1}^{K} \lambda^{\frac{1}{2}}(i) \underline{c}_r \underline{r}_i^T \tag{5-10b}$$

Substituting Eq. (5-8) into Eq. (5-10) yields in Eq. (5-11),

$$\alpha = m_f[1.0 - \sum_i \sum_j \hat{H}_{SVD}(i,j)] \tag{5-11}$$

where α represents the mean difference, i.e., $m_g - m_{\hat{g}_{SVD}}$.

In order for $m_{\hat{g}_{SVD}}$ to be equal to m_g, we simply add the quantity α to every output pixel. This simple point by point operation will significantly reduce the ε_1 error. Hence, the overall error ε_3 will be reduced.

Table 5-4 shows the effectiveness of the mean correction procedure in overall performance. Compare ε_3 and ε_4, where ε_4 denotes the total error after mean correction. The extra computation of m_f and $\sum_i \sum_j \hat{H}_{SVD}(i,j)$ would be fully justified in the bandpass filter case since a substantial reduction in NMSE can be obtained. The usefulness of this simple mean correction procedure in a photographic example will be demonstrated further in the next section.

TABLE 5-4

The NMSE comparison of before and after mean correction

N	8	10	12	14	16	No Rounding
Before	1.3445	0.3174	0.0883	0.0297	0.0249	0.0243
After	1.3449	0.3180	0.0864	0.0265	0.0179	0.0173

Lowpass Filter

N	8	10	12	14	16	No Rounding
Before	7.9674	4.0326	3.5160	3.4989	3.4984	3.4981
After	7.3556	2.0835	1.1237	1.0170	1.0156	1.0143

Bandpass Filter

The experiments have been repeated with varying correlation coefficients of input arrays. The results are shown in Table 5-5. Experimentally, it has been concluded that the fixed-point implementation error is quite independent of the correlation coefficient of the input array. These results confirm that the model employed was sufficiently valid for simulation.

5.3 Real Image Experimental Results

In this section, photograpic results, based on computer simulation, for SVD/SCK convolution are presented. The SVD/SCK convolution method with a fixed-point arithmetic has been applied to the convolution of real images as a test of its validity.

From previous experimental results, using random number arrays as an input, it was concluded that 16 bits of word-length for filter coefficient quantization and 12 bits for data storage, i.e., rounding, were sufficient to limit the effects of quantization and roundoff noise to less than 1.0 % NMSE for most practical cases. Although this conclusion is based on the particular model discussed in the previous section, we shall use the same word-length in the experiment with real images. Figure 5-5a shows an original aerial scene image. The original image contains 256x256 pixels with each pixel amplitude quantized over the integer range 0 to 255. In the first step of the

TABLE 5-5

Summary of experiment with varying correlation coefficient of the input array

ε_i \ ρ	0.0	0.1	0.3	0.5	0.7	0.9
ε_1	0.0807	0.0771	0.0684	0.0573	0.0442	0.0289
ε_2	0.1227	0.1175	0.1224	0.1144	0.1180	0.1072
ε_3	0.1463	0.1391	0.1369	0.1304	0.1273	0.1187
ε_4	0.1462	0.1393	0.1391	0.1273	0.1252	0.1195

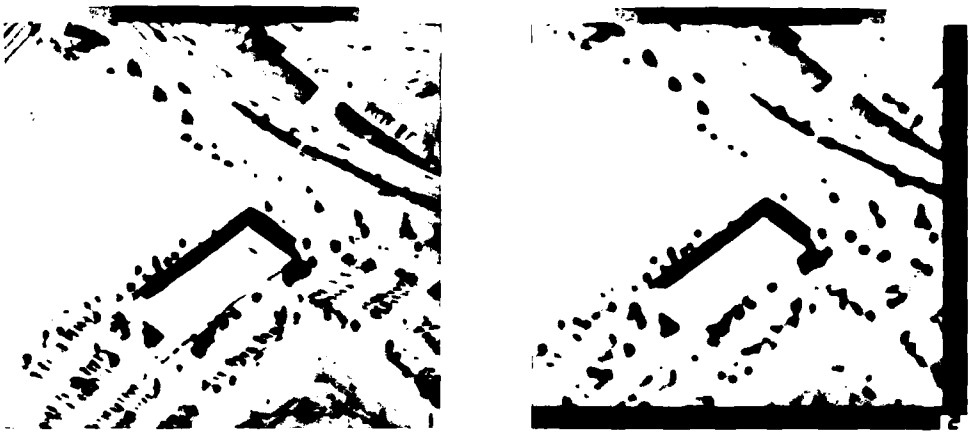
Lowpass Filter

ε_i \ ρ	0.0	0.1	0.3	0.5	0.7	0.9
ε_1	2.9375	2.9332	2.9609	3.0376	3.1653	3.4213
ε_2	0.4248	0.4201	0.4288	0.4369	0.4485	0.4483
ε_3	2.9557	2.9335	2.9681	3.050	3.1625	3.4440
ε_4	0.7460	0.7562	0.7561	0.7967	0.8378	1.0970

Bandpass Filter

simulation, each pixel of the original image was normalized to the range 0.0 to 1.0. Figures 5-5b and 5-5c illustrate the direct processed output with prototype lowpass and bandpass filters, respectively. The direct processed outputs were obtained using floating-point arithmetic with 36 bits of word-length. A comparison of direct and SVD/SCK convolution for lowpass and bandpass filters with $N = 12$ and $M = 16$ bits, is given in Figures 5-6 and 5-7, respectively. There are no apparent differences in visual results for direct and SVD/SCK convolution. The measured NMSE and absolute difference image, multiplied by a specified scale factor, are also presented to show the accuracy of SVD/SCK convolution. In both cases, the resulting errors are less than 1.0 %. This experiment verifies the validity of the model used in the previous section. Figures 5-8 and 5-9 contain simulation results for the experiment of Figures 5-6 and 5-7 when the word-length for data storage is reduced by setting $N = 8$. Obviously, the error contribution caused by insufficient word-length for rounding is significant.

To illustrate how the different SVD approximations of a given prototype impulse response affect the outputs, Figures 5-10 and 5-11 show the SVD/SCK processed output with different K. In this experiment, $N = 12$ and $M = 16$ were assumed. It is noted that the filter with $K = 1$ corresponds to the MMSE separable approximation of the



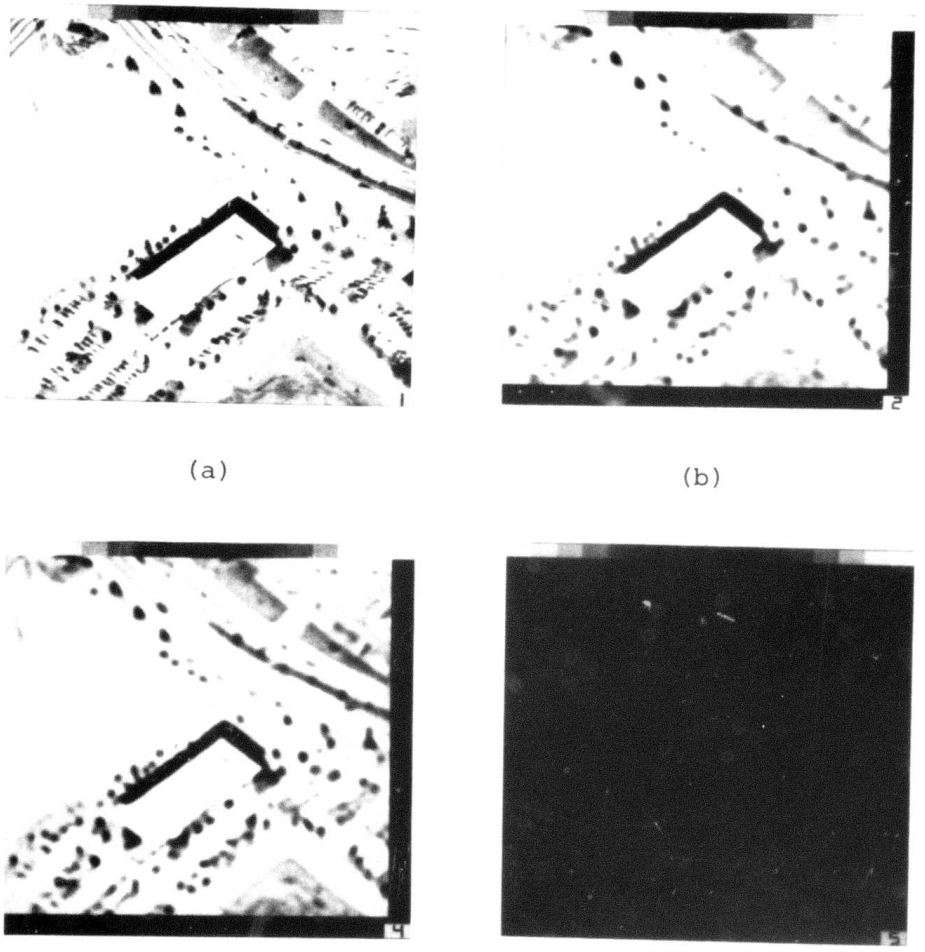
(a) (b)

(c)

Figure 5-5. Example of direct processing convolution

a) Original
b) Lowpass filter
c) bandpass filter

Best
Available
Copy

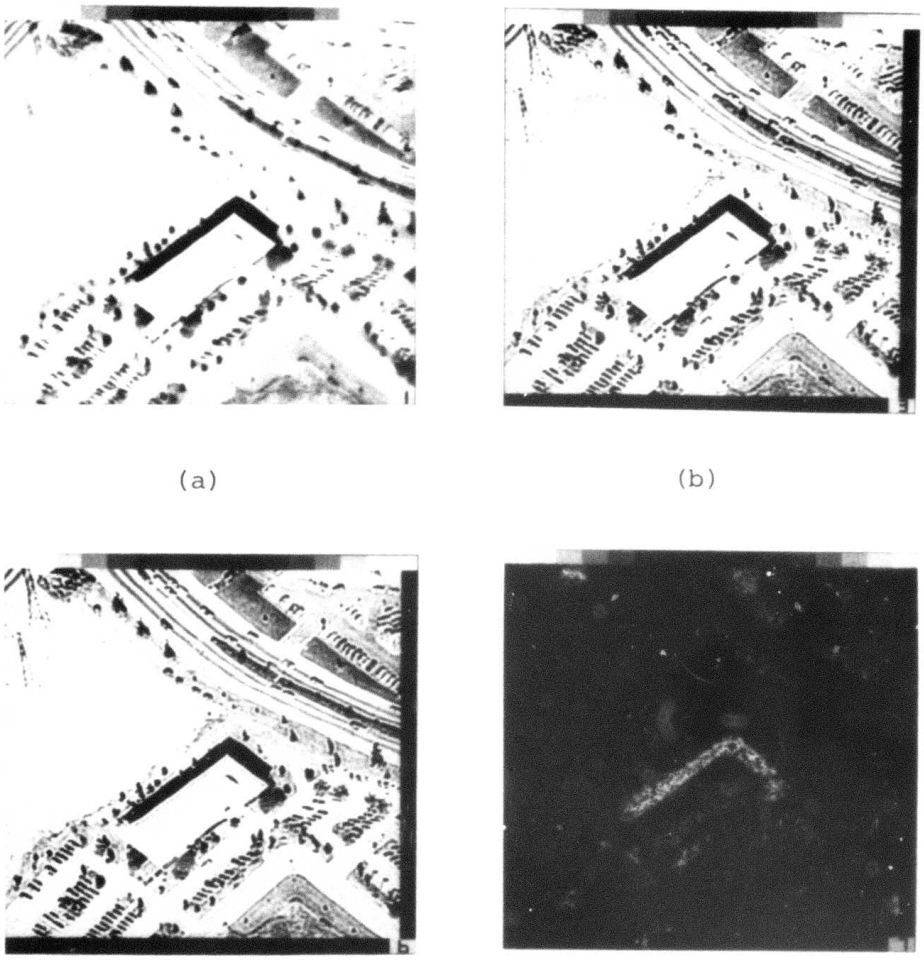


(a) (b)

(c) (d)

Figure 5-6. Comparison of direct and SVD/SGK Convolution for lowpass filter with L=15, K=3, M=16 bits and N=12 bits.

a) Original
b) Direct
c) SVD/SGK (NMSE=0.06398%)
d) Absolute difference X scale factor 200

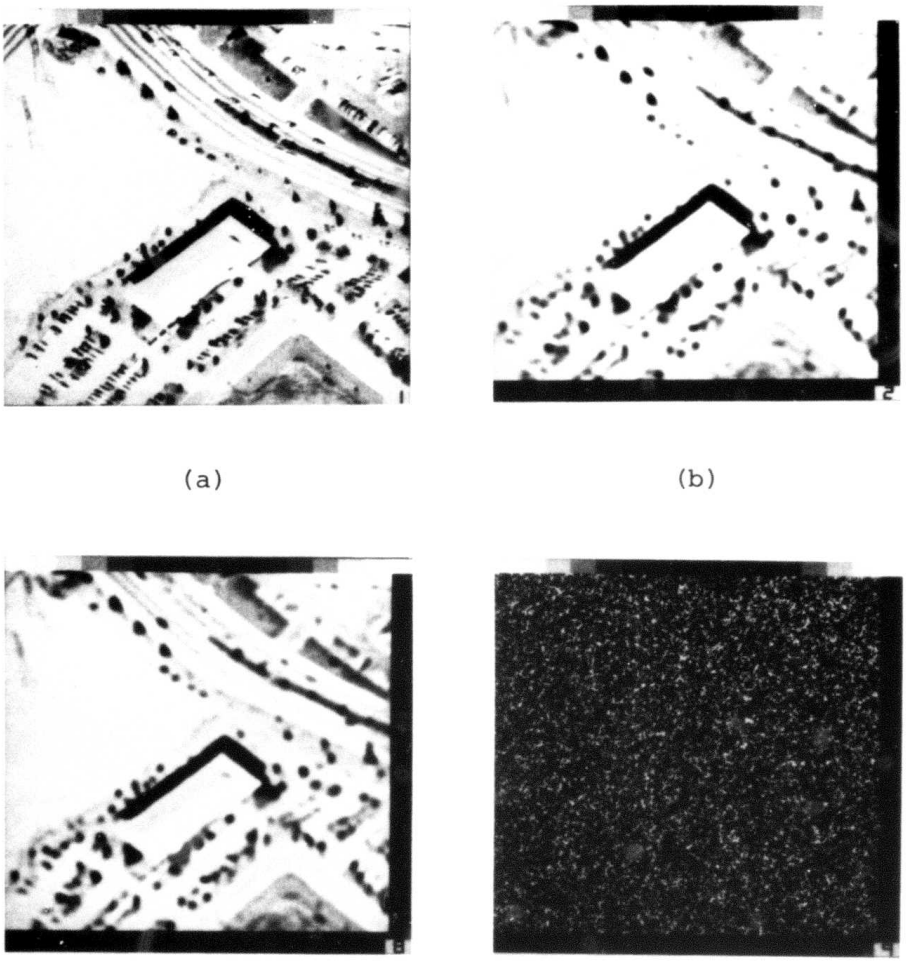


(a) (b)

(c) (d)

Figure 5-7. Comparison of direct and SVD/SGK convolution for bandpass filter with L=11, K=4, M=16 bits and N=12 bits

a) Original
b) Direct
c) SVD/SGK (NMSE=0.8742%)
d) Absolute difference X 40

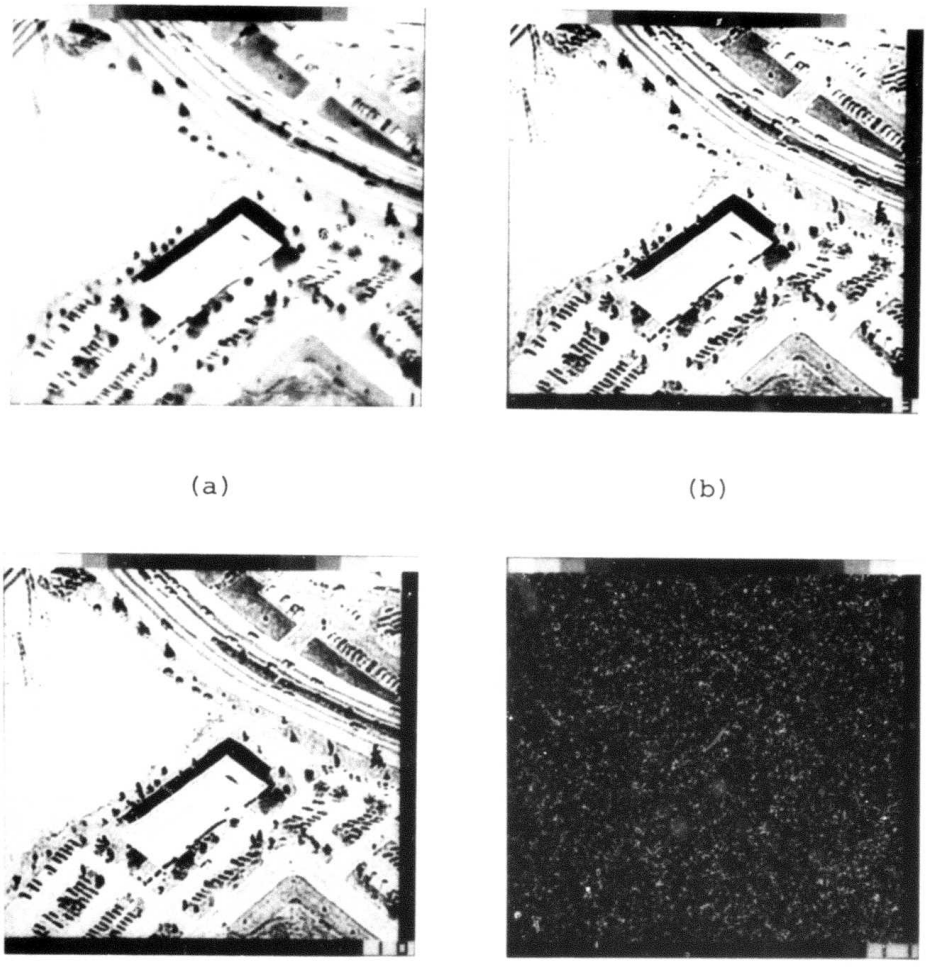


(a) (b)

(c) (d)

Figure 5-8. Comparison of direct and SVD/SGK convolution for lowpass filter with L=15, K=3, M=16 bits and N=8 bits.

a) Original
b) Direct
c) SVD/SGK (NMSE=1.1037%)
d) Absolute difference X 200

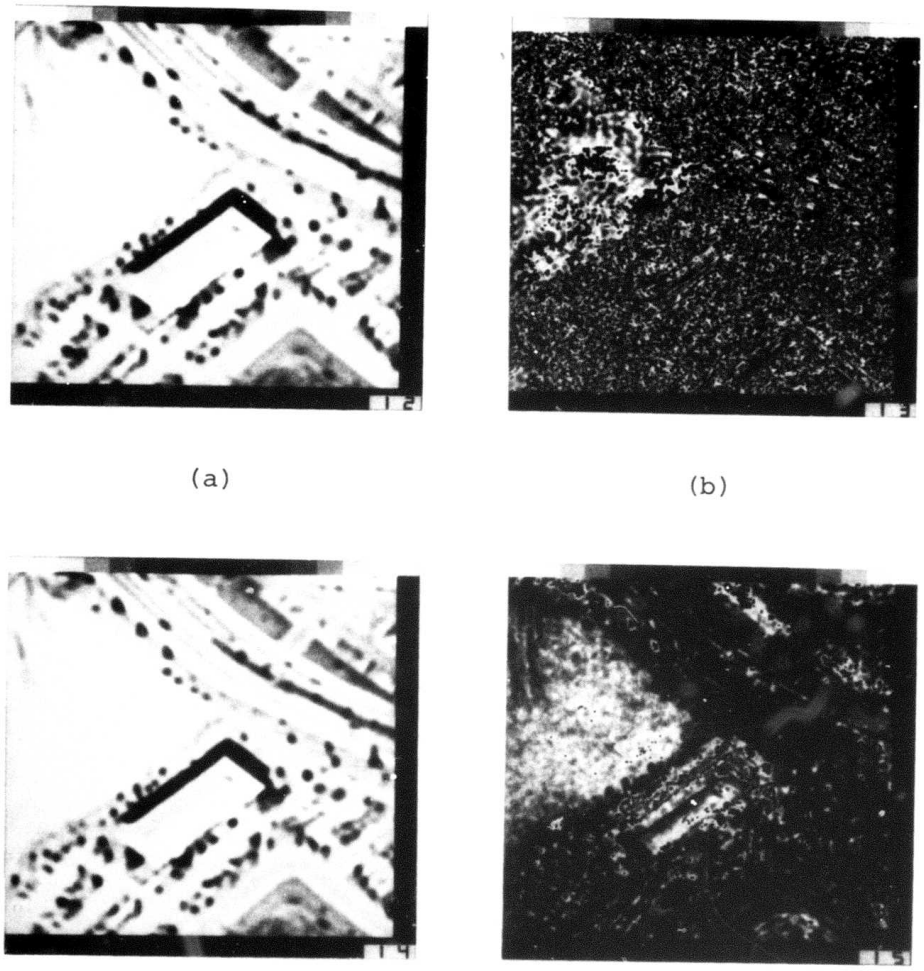


(a) (b)

(c) (d)

Figure 5-9. Comparison of direct and SVD/SGK convolution for bandpass filter with L=11, K=4, M=16 bits and N=8 bits.

a) Original
b) Direct
c) SVD/SGK (NMSE=8.049%)
d) Absolute difference X 40



(a) (b)

(c) (d)

Figure 5-10. Lowpass SVD/SGK convolution with K=1,2, L=15, M=16 bits and N=12 bits.

a) SVD/SGK, K=1
b) Absolute difference X 200
c) SVD/SGK, K=2
d) Absolute difference X 200

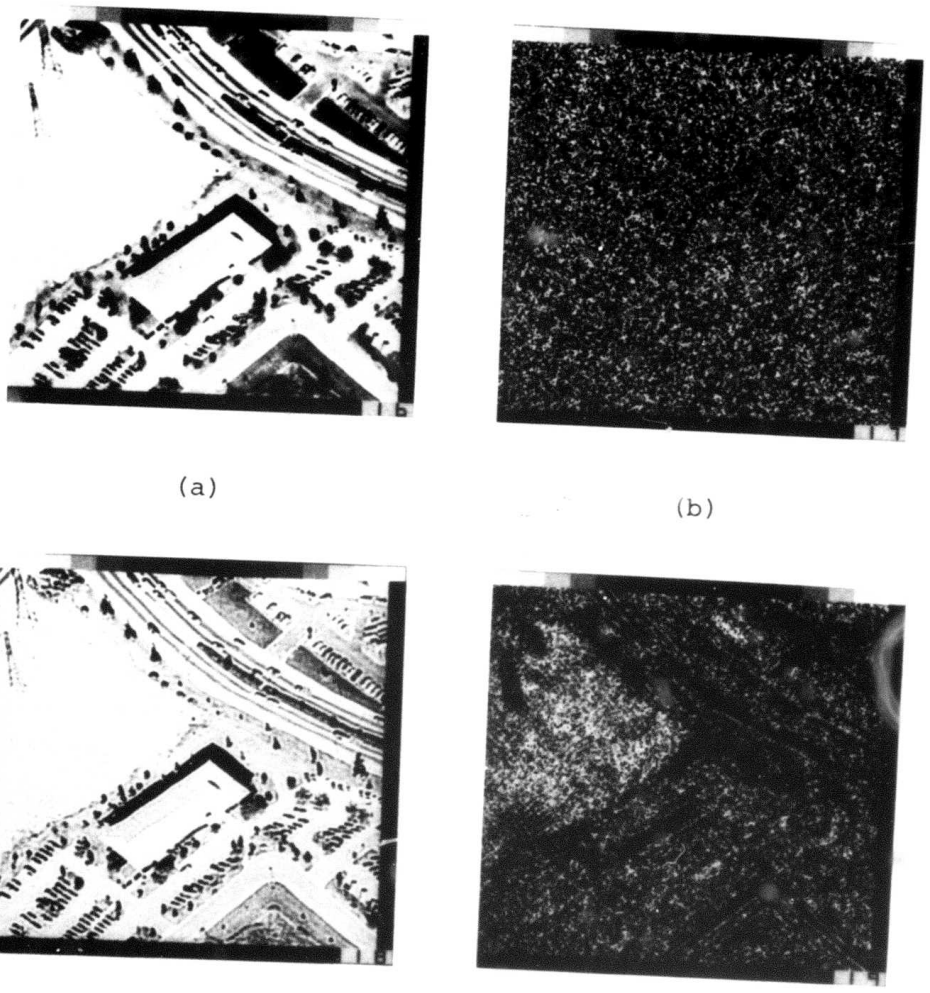
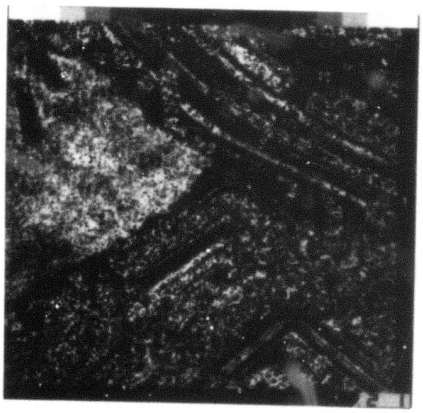


Figure 5-11. Bandpass SVD/SGK convolution with K=1,2,3, L=11, M=16 bits and N=12 bits.

a) SVD/SGK, K=1
b) Absolute difference X 40
c) SVD/SGK, K=2
d) Absolute difference X 40
e) SVD/SGK, K=3
f) Absolute difference X 40

(e)



(f)

Figure 5-11. (Continued)

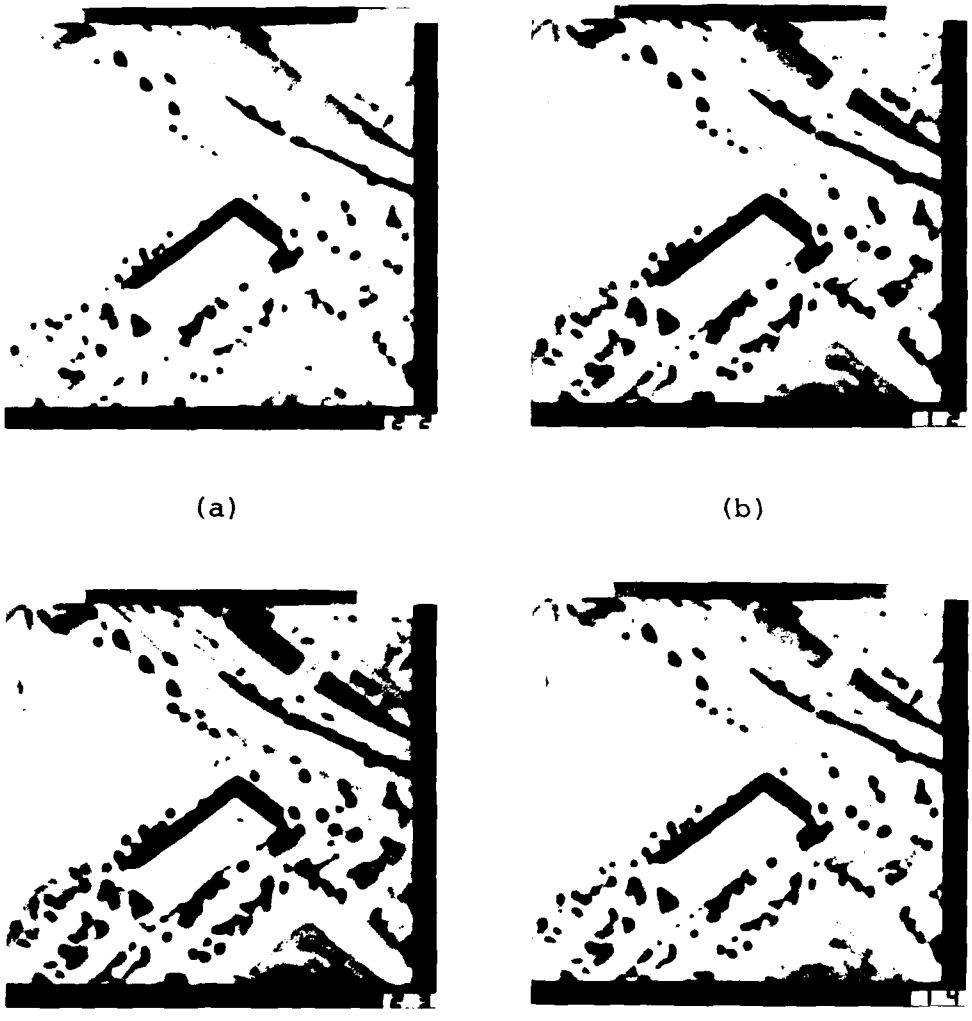
prototype impulse response. For the lowpass filter, there is no significant visual difference among different K's. But, there is a significant difference in the bandpass filter.

Figures 5-12 and 5-13 illustrate the effect of the mean correction algorithm. Although there is an obvious improvement in image quality in the bandpass filter, the improvement in the lowpass filter is not noticeable because the output mean before mean correction in the lowpass filter is already close to the input mean. The measured NMSEs (before and after), computed means, and $\sum_i \sum_j \hat{H}_{SVD}(i,j)$ are listed in Table 5-6.

For the bandpass filter with K = 1, before mean correction, the SVD approximation error is so severe that the SVD/SGK processed output is almost saturated. After mean correction, the output is subjectively satisfying, and the resulting NMSE is significantly reduced. This experiment visually demonstrates the effectiveness of the mean correction procedure.

5.4 Conclusion

This chapter has presented experimental results of SVD/SGK convolution using fixed-point arithmetic based on computer simulation. First, the derived noise formula predicting roundoff noise has been confirmed

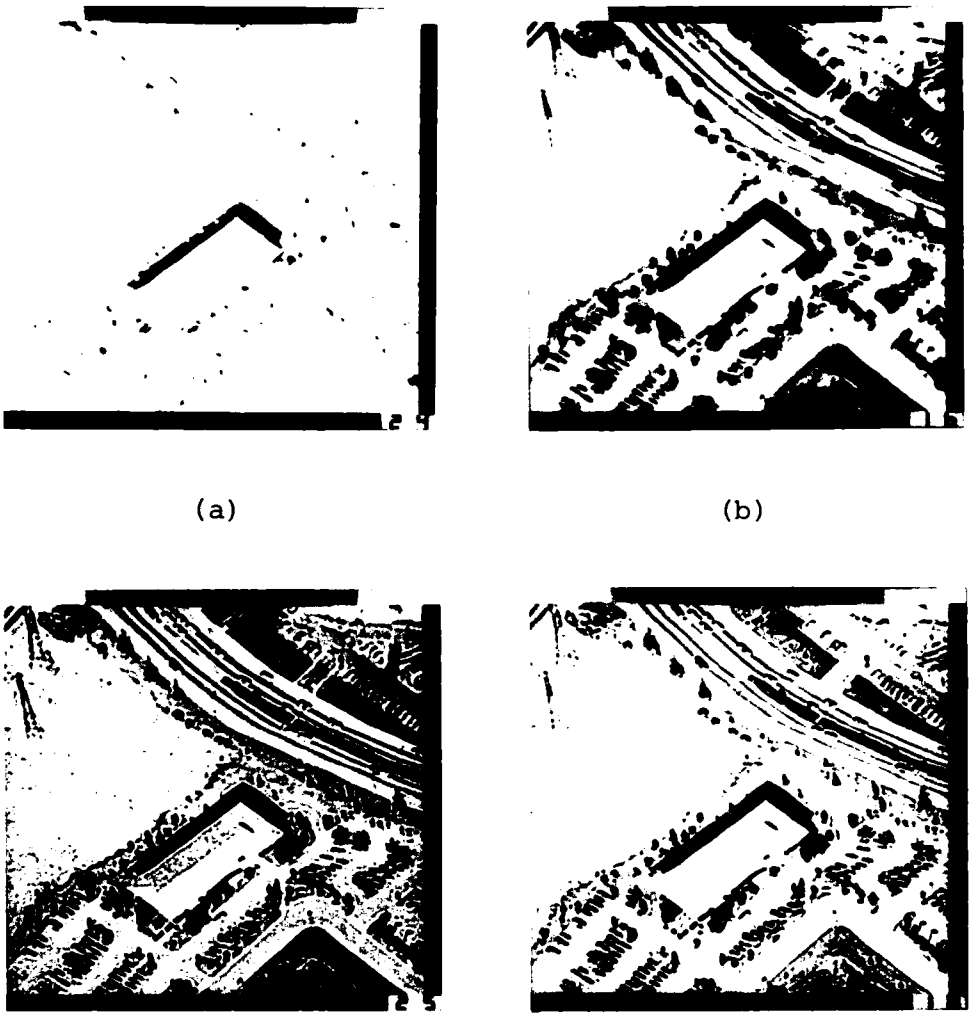


(a) (b)

(c) (d)

Figure 5-12. Comparison between before and after mean correction with L=15, M=16 bits and N=12 bits (lowpass).

a) SVD/SGK (K=1), before
b) SVD/SGK (K=1), after
c) SVD/SGK (K=2), before
d) SVD/SGK (K=2), after

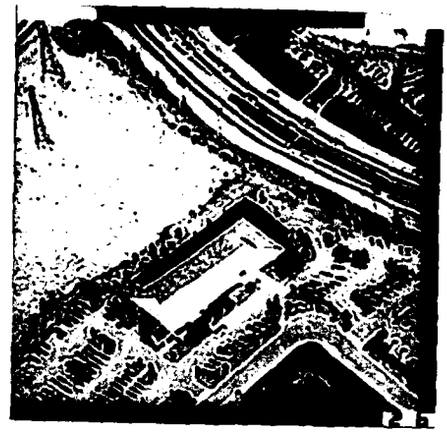


(a) (b)

(c) (d)

Figure 5-13. Comparison between before and after mean correction with L=11, M=16 bits, and N=12 bits (bandpass).

a) SVD/SGK (K=1), before
b) SVD/SGK (K=1), after
c) SVD/SGK (K=2), before
d) SVD/SGK (K=2), after
e) SVD/SGK (K=3), before
f) SVD/SGK (K=3), after



(e)

(f)

Figure 5-13 (Continued)

TABLE 5-6

Summary of experiment with real image

k	Before		After		$\sum_i \sum_j \hat{H}_{SVD}(i,j)$
	NMSE(%)	Mean	NMSE(%)	Mean	
1	12.6752	0.8122	1.9035	0.7210	1.1278
2	3.1634	0.6973	0.3658	0.72006	0.9682
3	0.0739	0.7201	0.0640	0.72001	0.9998

Lowpass Filter $m_f = 0.7202064$

k	Before		After		$\sum_i \sum_j \hat{H}_{SVD}(i,j)$
	NMSE(%)	Mean	NMSE(%)	Mean	
1	134.6718	1.7231	23.3312	0.7202	2.3928
2	15.5961	0.6046	3.3338	0.7199	0.8397
3	13.8732	0.6167	2.5028	0.7200	0.8564
4	2.8910	0.7592	0.8742	0.7206	1.0543

Bandpass Filter $m_f = 0.7202064$

experimentally. This experiment verifies that the statistical noise model used to analyze the roundoff error. It has been found that M = 16 bits and N = 12 bits are sufficient to limit the effects of quantization and roundoff noise to less than 1.0 % NMSE in most cases. To obtain a reasonable decrease in the NMSE, a simple mean correction algorithm was proposed. The image quality improvement obtainable by resetting the output mean equal to the input mean has been demonstrated. The pictorial images resulting from SVD/SCK convolution, as shown in this chapter, suggest that this technique may have some application in real-time image display system.

CHAPTER 6

PARAMETRIC DESIGN AND FIXED-POINT IMPLEMENTATION OF SVD/SGK CONVOLUTION FILTERS

6.1 Introduction

In this chapter, we will consider the problem of designing an SVD/SGK convolution filter for which the cutoff frequency is parametrically variable. Variable cutoff frequency filters have numerous applications in image processing. For example, one might sequentially obtain a best restored image by changing the cutoff frequency, hence the frequency response, of the restoration operator.

Since filter coefficients are generally a function of the filter cutoff frequency, one can change the filter cutoff frequency by varying all of the filter coefficients. But this procedure requires changing a number of parameters. Therefore, it is often impractical and too complicated. It would be more practical if one could construct a filter so that the cutoff frequency is controlled by only a few parameters, say one or two.

Based on the earlier work of Constandinides [6-1,6-2], Schussler and Winkelnkemper [6-3] were the first to design such a variable cutoff frequency digital filter. It has been shown, that by replacing each delay element in the basic filter structure with a first-order all-pass network, a transformed filter whose frequency response is identical to that of the basic filter on a distorted frequency scale is obtained. Unfortunately, the method described is restricted to FIR filters, and is not applicable to IIR filters. Furthermore, the resulting transformed filter is an IIR filter because by replacing the basic element, the first-order all-pass network becomes recursive. Consequently, the linear phase property of the basic FIR filter is lost. But, the variation of the cutoff frequency can be accomplished. Oppenheim et al. [6-4] proposed a new frequency transformation technique in which the resulting transformed filter is still an FIR filter, and the phase is linear if the basic filter is a linear phase FIR filter. By noting the fact that the SVD/SGK convolution filter is essentially a sum of separable filters, each weighted by a singular value, and each separable filter is an outer product of one-dimensional column and row convolution operators, it is possible to extend the proposed one-dimensional frequency transformation technique to the SVD/SGK convolution filters. We shall show that this approach is quite successful in designing a variable cutoff

SVD/SGK convolution filter. In this chapter, we shall discuss only a lowpass-to-lowpass transformation. Modification for highpass-to-highpass or bandpass-to-bandpass is rather straightforward in most cases. We assume here that the basic filter is a two-dimensional FIR filter with linear phase. The basic concepts of frequency transformation and modification to the SVD/SGK convolution filter are discussed in Section 6-2. A fixed-point implementation of the variable cutoff SVD/SGK convolution filter and experimental results are described in Section 6-3.

6.2 Frequency Transformation of Linear Phase FIR Filters

A one-dimensional FIR filter with impulse response of length 2Q+1 has a frequency response

$$h(e^{ju}) = \sum_{m=0}^{2Q} h(m)e^{-jmu} \qquad (6\text{-}1)$$

A linear phase filter is symmetrical so that

$$h(m) = h(Q-m) \qquad (6\text{-}2)$$

for $m = 0,1,\ldots,Q$. Thus

$$h(e^{ju}) = e^{-juQ}[h(Q) + \sum_{m=1}^{Q-1} 2h(m)\cos[u(Q-m)]] \qquad (6\text{-}3)$$

Letting n = Q-m, Eq. (6-3) becomes

$$h(e^{ju}) = e^{-juQ} \sum_{n=0}^{Q} a(n)\cos u \qquad (6-4)$$

where $a(0) = h(Q)$ and $a(n) = 2h(Q-n)$ for $n = 1,2,\ldots,Q$.

We note that

$$T_n(\cos u) = \cos nu \qquad (6-5)$$

where T_n is the n-th degree Chebyshev polynomial that satisfies the recursion formula

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \qquad (6-6)$$

for $n = 1,2,\ldots,Q$. Thus, Eq. (6-4) can be reformulated as

$$h(e^{ju}) = e^{-juQ} \sum_{n=0}^{Q} b(n)(\cos u)^n \qquad (6-7)$$

The new coefficient b(n), for $n = 0,1,\ldots,Q$, is obtained from the Chebyshev polynomial recursion formula of Eq. (6-6). The basic approach [6-5] to the variable cutoff linear phase filter is to use the transformation

$$\cos u = \sum_{k=0}^{P} A_k(\cos\beta)^k \qquad (6-8)$$

where u and β are the frequency variables of the basic and transformed filters, respectively. The transformation described above preserves the frequency response of the

basic filter, although the frequency scale is distorted by the transformation. By substituting the transformation of Eq. (6-8) into Eq. (6-7), the frequency response of the transformed filter is found to be

$$h_T(e^{j\beta}) = e^{-j\beta QP}\left[\sum_{n=0}^{Q} b(n) \sum_{k=0}^{P} A_k(\cos\beta)^k\right] \qquad (6\text{-}9)$$

From Eq. (6-9), it is noted that the impulse response dimension of the transformed filter is now 2QP+1. By appropriately controlling the parameters A_k, for $k = 0,1,\ldots,P$, the cutoff frequency of the transformed filter can be varied.

If P = 1, then Eq. (6-8) becomes

$$\cos u = A_0 + A_1\cos\beta \qquad (6\text{-}10)$$

and for P = 2, the transformation assumes the form

$$\cos u = A_0 + A_1\cos\beta + A_2\cos^2\beta \qquad (6\text{-}11)$$

We shall call the transformations of Eq. (6-10) and Eq. (6-11) first-order and second-order transformations, respectively. The nature of the first-order transformation is depicted in Fig. 6-1. For first-order transformation, if one is interested in increasing the cutoff frequency, i.e., $u_c \leq \beta_c$, where u_c and β_c correspond to the cutoff frequency of the basic and transformed filters, respectively, one may prefer to constrain the transformed

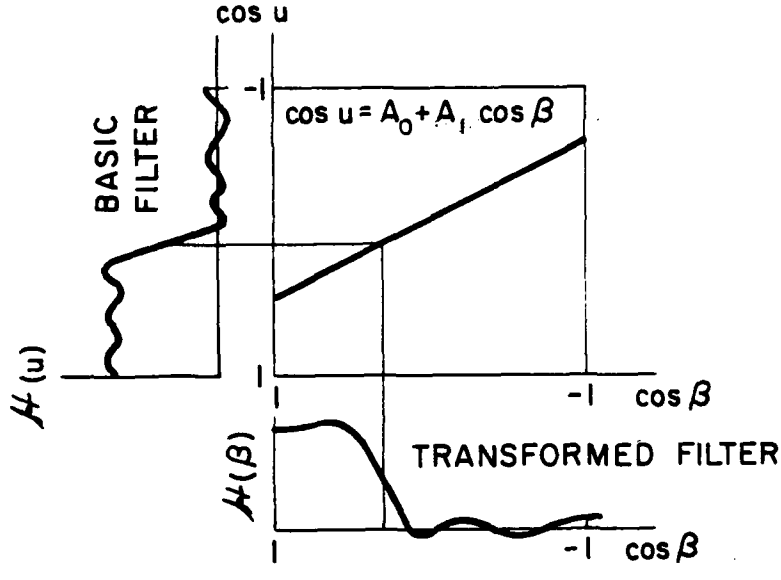


Figure 6-1. Nature of the first-order transformation

frequency response magnitude at $\beta = 0$ to be equal to that of the basic filter. Mathematically, it can be shown as

$$h_T(e^{j\beta})\Big|_{\beta=0} = h(e^{ju})\Big|_{u=0} \tag{6-12}$$

in which case, $A_0+A_1 = 1$. But, in order to ensure that $|\cos u| \leq 1$, A_0 should lie in the range of

$$0 \leq A_0 < 1 \tag{6-13}$$

By changing the frequency control parameter A_0 from zero to unity, we can obtain a transformed filter whose cutoff frequency is given by

$$\beta_c = \cos^{-1}\left[\frac{\cos u_c - A_0}{1-A_0}\right] \tag{6-14}$$

In other words, if we wish to increase the filter cutoff frequency from u_c to β_c with $u_c \leq \beta_c$, then the control parameter A_0 is obtained from Eq. (6-15) as

$$A_0 = \frac{\cos\beta_c - \cos u_c}{\cos\beta_c - 1} \tag{6-15}$$

where $0 \leq A_0 < 1$. To decrease the cutoff frequency of the basic filter, the correspondence is

$$h_T(e^{j\beta})\Big|_{\beta=\pi} = h(e^{ju})\Big|_{u=\pi} \tag{6-16}$$

Equation (6-16) leads to the constraint that $A_1 = 1-A_0$; the parameter A_0 is restricted to the range of

$$-1 < A_0 \leq 0 \tag{6-17}$$

The resulting transformed filter cutoff frequency is given by

$$\beta_c = \cos^{-1}\left[\frac{\cos u_c - A_0}{1 + A_0}\right] \tag{6-18}$$

Let us associate the complex variable z with the basic filter system function H(z) and the complex variable Z with the transformed filter system function $H_T(Z)$. Then, the transformation of Eq. (6-8) is equivalent to

$$\frac{z+z^{-1}}{2} = \sum_{k=0}^{P} A_k \left(\frac{Z+Z^{-1}}{2}\right)^k \tag{6-19}$$

If the filter is implemented as a cascade of SGK filters, it is noted here that the SGK filter should be symmetrical because the transformation is applicable only to a linear phase filter. In Chapter 2, it was shown that the complex zeros of H(z) should be grouped together in conjugate pairs to ensure that all kernels for 3x1 SGK filters are real. But, a resulting 3x1 SGK filter may not be linear phase. For example, the 3x1 SGK filter from grouping complex conjugate pair zeros not on the unit circle will not be symmetrical. In order to ensure that all SGK filters are

linear phase, complex zeros not on the unit circle should be grouped together in groups of four, corresponding to the complex conjugates and reciprocals, i.e., α, α^*, $\frac{1}{\alpha}$, $\frac{1}{\alpha^*}$. As a consequence, H(z) will have fourth-order SGK filter with system function of the form

$$H_i(z) = 1-2\left(\frac{r_i^2+1}{r_i}\right)\cos\theta_i z^{-1}+\left(r_i^2+\frac{1}{r_i^2}+4\cos^2\theta_i\right)z^{-2} -2\left(\frac{r_i^2+1}{r_i}\right)\cos\theta_i z^{-3}+z^{-4} \tag{6-20}$$

where r_i and θ_i are the magnitude and phase of one of the complex zeros not on the unit circle.

The same rule of zero grouping in Chapter 2 can be applied to the real zeros and complex zeros on the unit circle. Therefore, we can obtain a realization of H(z) in terms of a cascade of second- or fourth-order linear phase SGK filters. The z-transform of the second- or fourth-order SGK filters can be written as

$$H_i(z) = \sum_{n=0}^{2} b_i(n)\left(\frac{z+z^{-1}}{2}\right)^n \tag{6-21}$$

But H(z) can be factored in the form

$$H(z) = \prod_{i=1}^{Q_1} H_i(z) = \prod_{i=1}^{Q_1}\left[\sum_{n=0}^{2} b_i(n)\left(\frac{z+z^{-1}}{2}\right)\right]^n \tag{6-22}$$

for $Q_1 \leq Q$. To obtain a variable cutoff linear phase filter, based on the SVD/SGK convolution, each SGK filter is transformed in the manner described earlier. The Z-transform of the transformed filter is

$$H_T(Z) = \prod_{i=1}^{Q_1} H_{T_i}(Z) = \prod_{i=1}^{Q_1} \left\{ \sum_{n=0}^{2} b(n) \left[\sum_{k=0}^{P} A_k \left(\frac{Z+Z^{-1}}{2} \right)^k \right] \right\} \quad (6\text{-}23)$$

Therefore, the coefficients of the transformed SGK filter are expressed in terms of the parameters A_k and the coefficients of the basic filter (see Appendix B). By controlling the parameter A_k, the transformed filter cutoff frequency can be varied. Before we present the experimental results, let us define a

$$R = \frac{\beta_c - u_c}{u_c} \times 100 \quad (6\text{-}24)$$

which will be used to describe the degree of the transformation. Figure 6-2 shows the frequency response of a typical lowpass filter and the parameters that define it. The three parameters Δ_1, Δ_2, and Δf characterize the frequency response of the filter. If the parameters for the transformed filter are close to those of the basic filter, then the transformation will adequately preserve the frequency response of the basic filter.

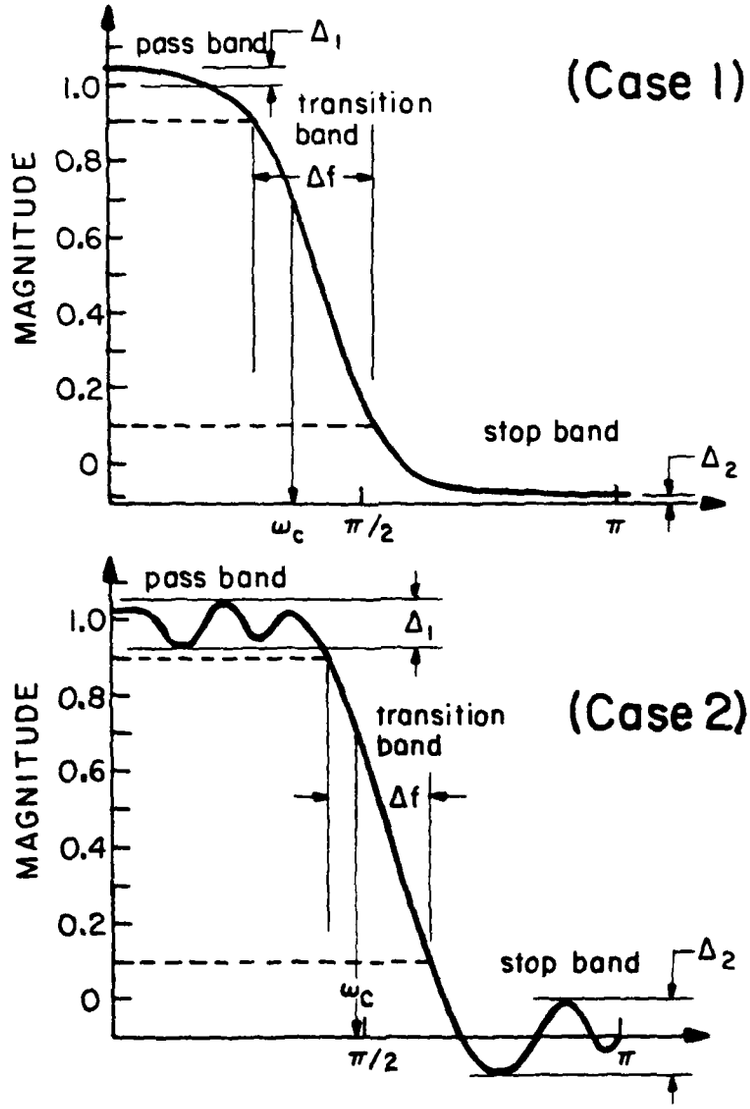


Figure 6-2. Definition of lowpass filter parameters

In order to verify the first-order transformation, the following experiment was performed. The basic filter is a one-dimensional linear phase lowpass filter with an impulse response length of 15. The measured cutoff frequency of the basic filter is 0.6739. Throughout this chapter, the specified frequency is normalized to the range of $(0,\pi)$. Figure 6-3 shows the frequency responses of the transformed filter with the parameter A_k varied from 0.1 to 0.8. Table 6-1 summarizes the filter parameters and the desired and measured cutoff frequencies of the transformed filters. There is excellent agreement between the two values. But it is observed that the first-order transformation does not adequately preserve the frequency response of the basic filter as A_0 goes to 1. In the case of $A_0 \geq 0.6$, the resulting transformed filters can not be considered to be lowpass filters, because the first-order transformation does not constrain the frequency response at $u = \beta = \pi$. Experimental evidence shows that there is a trade off between R and the preservation of the frequency response of the basic filter. To preserve the frequency response of the basic filter more adequately, R should be relatively small. Thus, the penalty paid for large R is that the transformed filter does not preserve the frequency response of the basic filter as shown in Fig. 6-3.

As alternative to the first-order transformation, one may apply a second-order transformation. In the

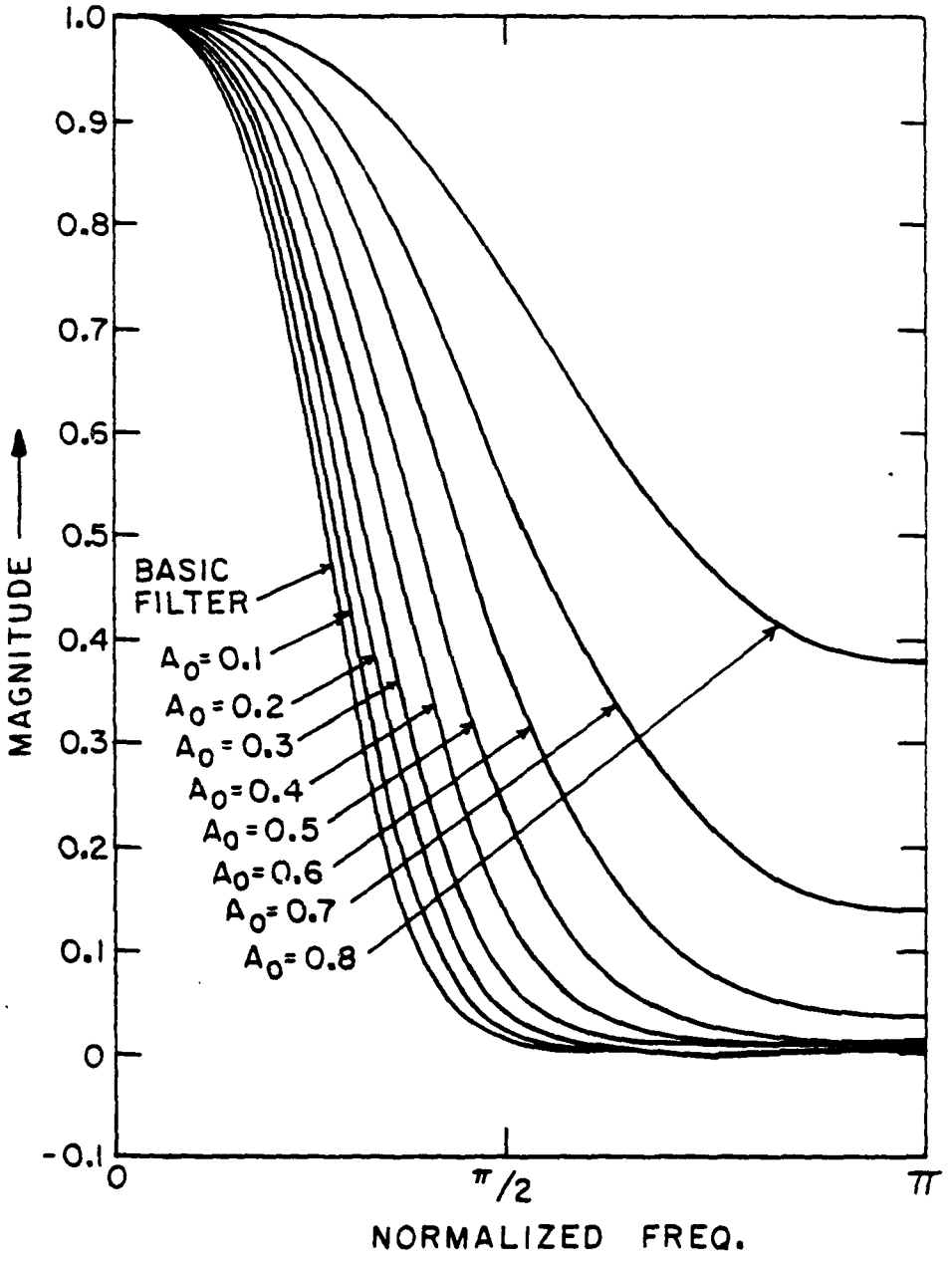


Figure 6-3. First-order transformation examples

TABLE 6-1

List of the transformed filters and their cutoff frequencies using first-order transformation

A_0	Δ_1	Δ_2	Δf	Cutoff Frequency		R(%)
				Desired	Measured	
0.0*	0.0	0.11×10^{-2}	0.367	0.6739	0.6739	0.0
0.1	0.0	0.39×10^{-3}	0.398	0.7119	0.7127	5.6
0.2	0.0	-0.22×10^{-2}	0.419	0.7572	0.7583	12.4
0.3	0.0	0.37×10^{-2}	0.471	0.8125	0.8137	20.6
0.4	0.0	0.10×10^{-1}	0.523	0.8819	0.8833	30.9
0.5	0.0	0.14×10^{-1}	0.701	0.9730	0.8571	44.4
0.6	0.0	0.38×10^{-1}	0.754	1.1001	1.1027	63.2
0.7	0.0	0.14×10^{0}	N.A.	1.2960	1.1299	92.3
0.8	0.0	0.38×10^{0}	N.A.	1.6640	1.6703	146.9

*A_0 = 0.0 means a basic filter.

second-order transformation, there are three parameters, A_0, A_1, and A_2 to be controlled. For the case in which the cutoff frequency of the transformed filter is greater than or equal to the cutoff frequency of the basic filter, we can put another constraint on the transformation. That is

$$h_T(e^{j\beta})\Big|_{\beta=\pi} = h(e^{ju})\Big|_{u=\pi} \tag{6-25}$$

By imposing the constraint of Eq. (6-25), we can preserve the frequency response of the basic filter better than when the first-order transformation is used. But we shall show that the second-order transformation severely restricts the range of transformation. By using a similar analysis, as used in the first-order transformation, it can be shown that the parameter A_0 is restricted to the range

$$\begin{aligned} 0 \le A_0 \le \frac{1}{2} \qquad & u_c \le \beta_c \\ -\frac{1}{2} \le A_0 < 0 \qquad & u_c > \beta_c \end{aligned} \tag{6-26}$$

and the desired cutoff frequency β_c is given by

$$\beta_c = \cos^{-1}\left[\frac{1-\sqrt{1-4A_0(\cos u_c - A_0)}}{2A_0}\right] \tag{6-27}$$

Detailed derivations of Eqs. (6-26) and (6-27) are given in Appendix C. Although the transformation is achieved by varying the parameter A_0, the maximum or minimum R can be obtained when $A_0 = 1/2$ or $-1/2$,

respectively. By substituting $A_0 = \pm 1/2$ into Eq. (6-27), we obtain the maximum or minimum attainable cutoff frequency with the second-order transformation. The relationship between u_c and maximum or minimum attainable β_c is shown in Fig. 6-4. Figure 6-5 shows the results with the second-order transformation. Table 6-2 shows the measured filter parameters. The basic filter is the same as previously described. Upon comparison of the first-order and the second-order transformations, the second-order transformation is seen to adequately preserve the frequency response of the basic filter if β_c is within the transform range. But, the resulting transformed filter has an impulse response length of $4Q+1$, instead of $2Q+1$ obtained as with the first-order transformation.

A second-order transformation is still possible even if the desired cutoff frequency β_c is out of the transform range. To extend the transform range, for the case of $u_c \leq \beta_c$, the constraint given in the Eq. (6-25) is forced to be satisfied at $\beta = \alpha_1$, where $0 \leq \alpha_1 \leq \pi$, rather than at $\beta = \pi$. The price paid for relaxing the constraint is that the transformed filter characteristics are sacrificed to some degree. In this case, the parameter A_0 lies in the range of

$$0 \leq A_0 \leq \frac{\cos\alpha_1 + 3}{4} \tag{6-28}$$

where the matching point α_1 is obtained from

$$\alpha_1 = \cos^{-1}\left[\frac{4(\cos u_c - 1)}{(\cos\beta_c - 1)^2} + 1\right] \tag{6-29}$$

But, it should be noted here that α_1 equals π whenever the desired β_c is within the transform range. For the case of $u_c > \beta_c$, the constraint given in Eq. (6-12) also can be relaxed by locating the matching point at $\beta = \alpha_2$, where $0 \leq \alpha_2 \leq \pi$. The range of A_0 can be shown to be

$$\frac{\cos\alpha_2 - 3}{4} \leq A_0 < 0 \tag{6-30a}$$

where

$$\alpha_2 = \cos^{-1}\left[\frac{4(\cos u_c + 1)}{(\cos\beta_c + 1)^2} - 1\right] \tag{6-30b}$$

Figure 6-6 illustrates the frequency response of the transformed filter with the (relaxed) second-order transformation. Table 6-3 lists the matching points, the desired and measured cutoff frequencies, and R. There is also good agreement between the desired and measured cutoff frequencies. As shown in Fig. 6-6, relaxation of the constraints in the second-order transformation allows the basic filter to transform in the desired manner, but the transformation gradually degrades the frequency response of

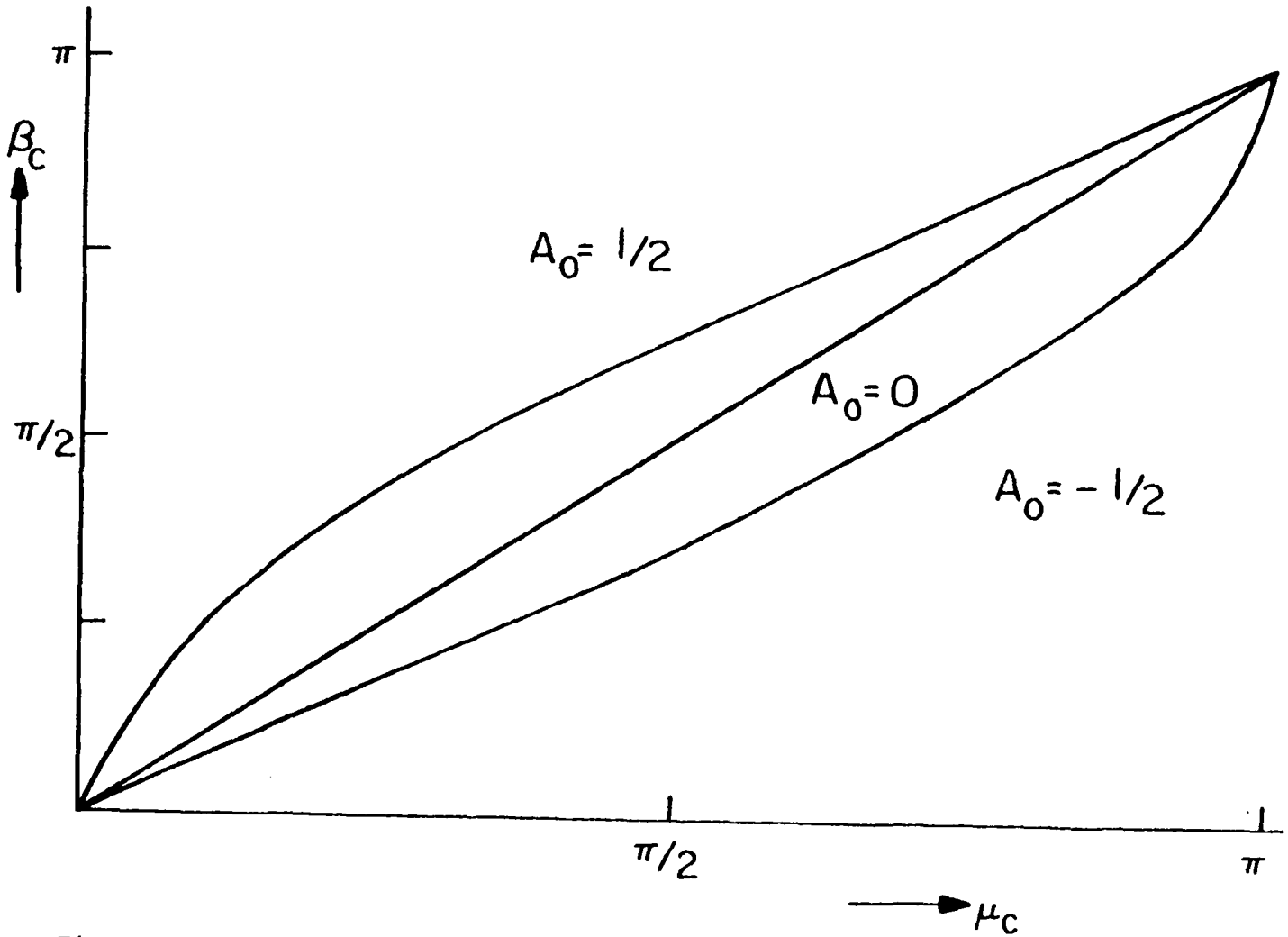


Figure 6-4. The transformation range with the second-order transformation

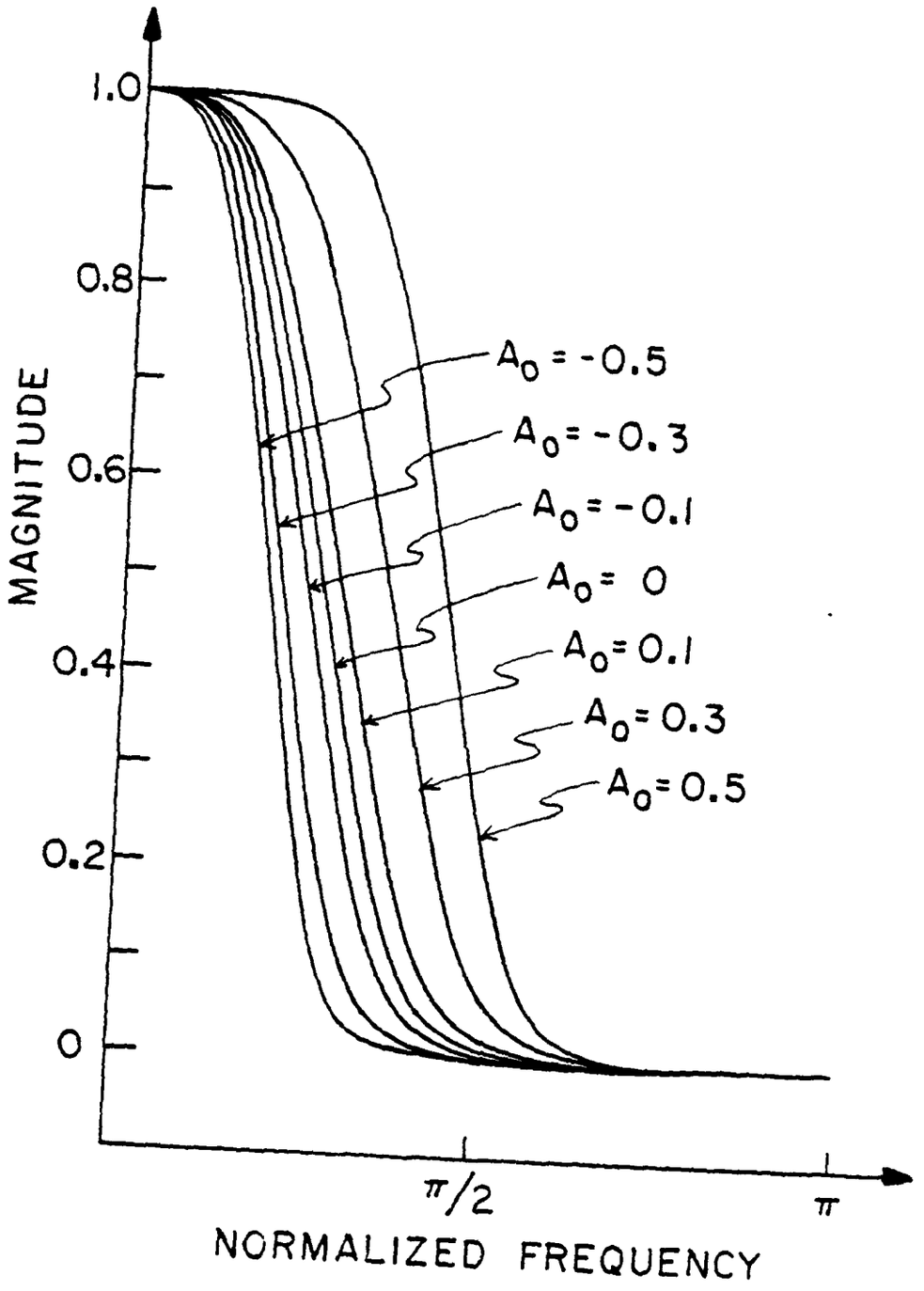


Figure 6-5. Second-order transformation examples

TABLE 6-2

List of the transformed filters and their cutoff frequencies using second-order transformation

A_0	Δ_1	Δ_2	Δf	Cutoff Frequency		R(%)
				Desired	Measured	
-0.5	0.0	0.11×10^{-2}	0.272	0.4788	0.4794	-28.9
-0.3	0.0	0.11×10^{-2}	0.304	0.5362	0.5371	-20.4
-0.1	0.0	0.11×10^{-2}	0.346	0.6181	0.6194	-8.3
0.0	0.0	0.11×10^{-2}	0.367	0.6739	0.6739	0.0
0.1	0.0	0.11×10^{-2}	0.398	0.7444	0.7463	10.5
0.3	0.0	0.11×10^{-2}	0.408	0.9477	0.9489	40.6
0.5	0.0	0.11×10^{-2}	0.356	1.2252	1.2283	81.8

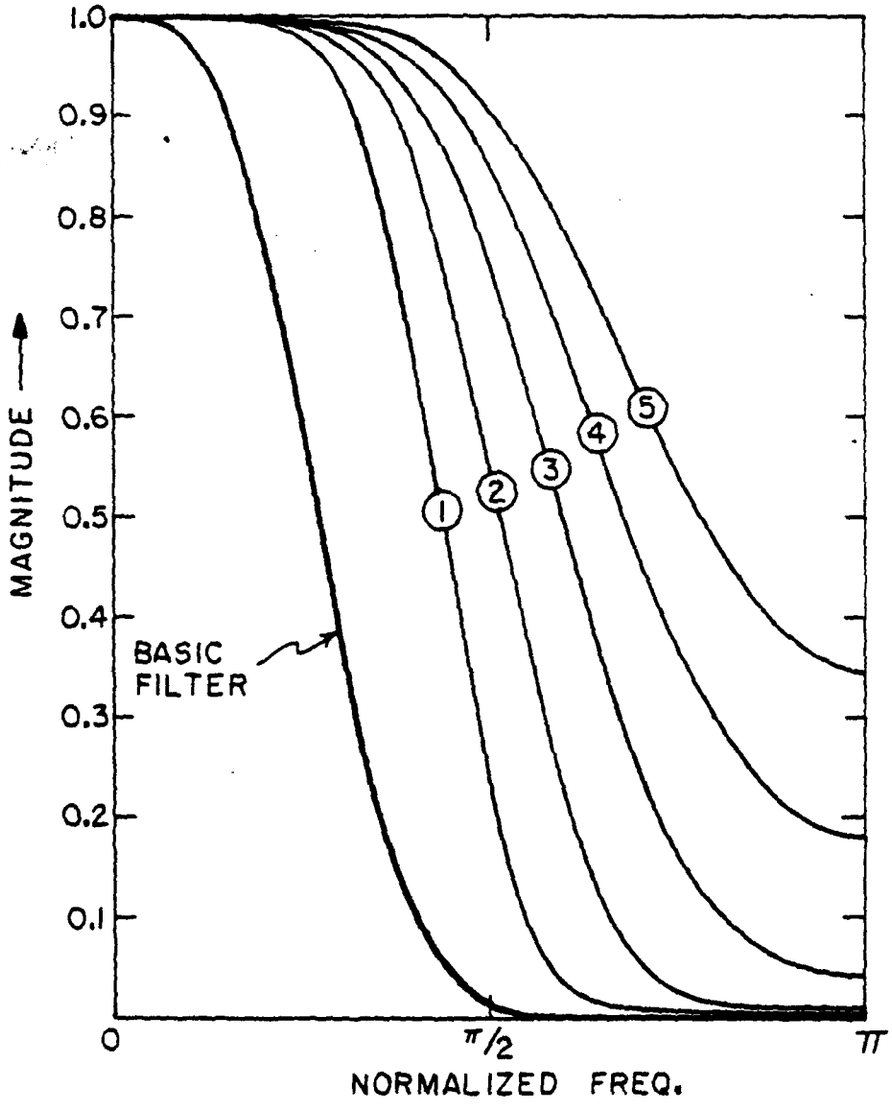


Figure 6-6. Second-order transformation examples with a relaxed constraint

TABLE 6-3

List of the transformed filters and their cutoff frequencies using (relaxed) second-order transformation

Filter	A_0	Matching Point α_1	Δ_1	Δ_2	Δf	Cutoff Frequency		R(%)
						Desired	Measured	
Basic Filter			0.0	0.11×10^{-2}	0.367	0.6739	0.6739	
1	0.5	π	0.0	0.11×10^{-2}	0.372	1.2252	1.2247	81.7
2	0.7009	1.7685	0.0	0.10×10^{-1}	0.471	1.4252	1.4256	111.6
3	0.8034	1.3559	0.0	0.41×10^{-1}	0.607	1.6252	1.6267	141.4
4	0.8605	1.1131	0.0	0.18×10^{0}	N.A.	1.8252	1.8280	171.3
5	0.8944	0.9549	0.0	0.34×10^{0}	N.A.	2.0252	2.0295	226.9

the transformed filter as R increases. It is believed that with the second-order transformation, there should also be a trade off between R and the preservation of the frequency response of the basic filter.

To extend the discussed frequency transformation technique to SVD/SGK convolution filters, it is noted again here that a SVD/SGK convolution filter is a sum of separable filters, and each separable filter is decomposed into an outer product of one-dimensional convolution operators on the columns and rows of the input image. If the basic SVD/SGK convolution filter is a two-dimensional FIR filter with linear phase, each SVD-expanded separable filter is also a two-dimensional FIR filter with linear phase. We assume here that the size of the basic filter is (2Q+1)x(2Q+1). Thus, each separable filter has the property that

$$H_i(n,m) = H_i(Q-n,Q-m), \quad 0 \leq m,n \leq Q \tag{6-31}$$

Since $H_i(n,m)$ is separable, then

$$H_i(n,m) = h_i^c(n)h_i^r(m) \tag{6-32}$$

But, from Eq. (6-31), $H_i(n,m)$ is also decomposed into

$$H_i(n,m) = h_i^c(Q-n)h_i^r(Q-m) \tag{6-33}$$

Therefore, the convolution operators on the column and row, $h_i^c(n)$ and $h_i^r(m)$, are also one-dimensional linear phase FIR

filters.

Figure 6-7 illustrates the frequency responses of the SVD-expanded separable filters on the horizontal axis. The prototype filter is a two-dimensional linear phase lowpass filter, and the SVD/SGK convolution filter was obtained by truncating the SVD expansion to 4 terms. The first-stage separable filter corresponding to the largest singular value shows almost the same frequency characteristics as the basic filter. But the other separable filters corresponding to the next largest singular values no longer are lowpass filters.

But a variable cutoff SVD/SGK convolution filter is obtained by simply transforming each of the one-dimensional convolution operators on the columns and rows of the input image. Specifically, the z-transform of the SVD/SGK convolution filter is given by

$$H_{SVD/SGK}(z_1,z_2)=\sum_{\ell=1}^{K}\left[\prod_{i=1}^{Q_1} H^{c}_{\ell,i}(z_1)\prod_{j=1}^{Q_2} H^{r}_{\ell,j}(z_2)\right] \qquad (6\text{-}34a)$$

or

$$H_{SVD/SGK}(z_1,z_2)=\sum_{\ell=1}^{K}\left\{\prod_{i=1}^{Q_1}\left[\sum_{n=1}^{2} b^{c}_{\ell,i}(n)\left(\frac{z_1+z_1^{-1}}{2}\right)^{n}\right]\cdot\prod_{i=1}^{Q_2}\left[\sum_{m=0}^{2} b^{r}_{\ell,i}(m)\left(\frac{z_2+z_2^{-1}}{2}\right)^{m}\right]\right\} \qquad (6\text{-}34b)$$

but $Q_1, Q_2 \le Q$. The Z-transform of the transformed SVD/SGK

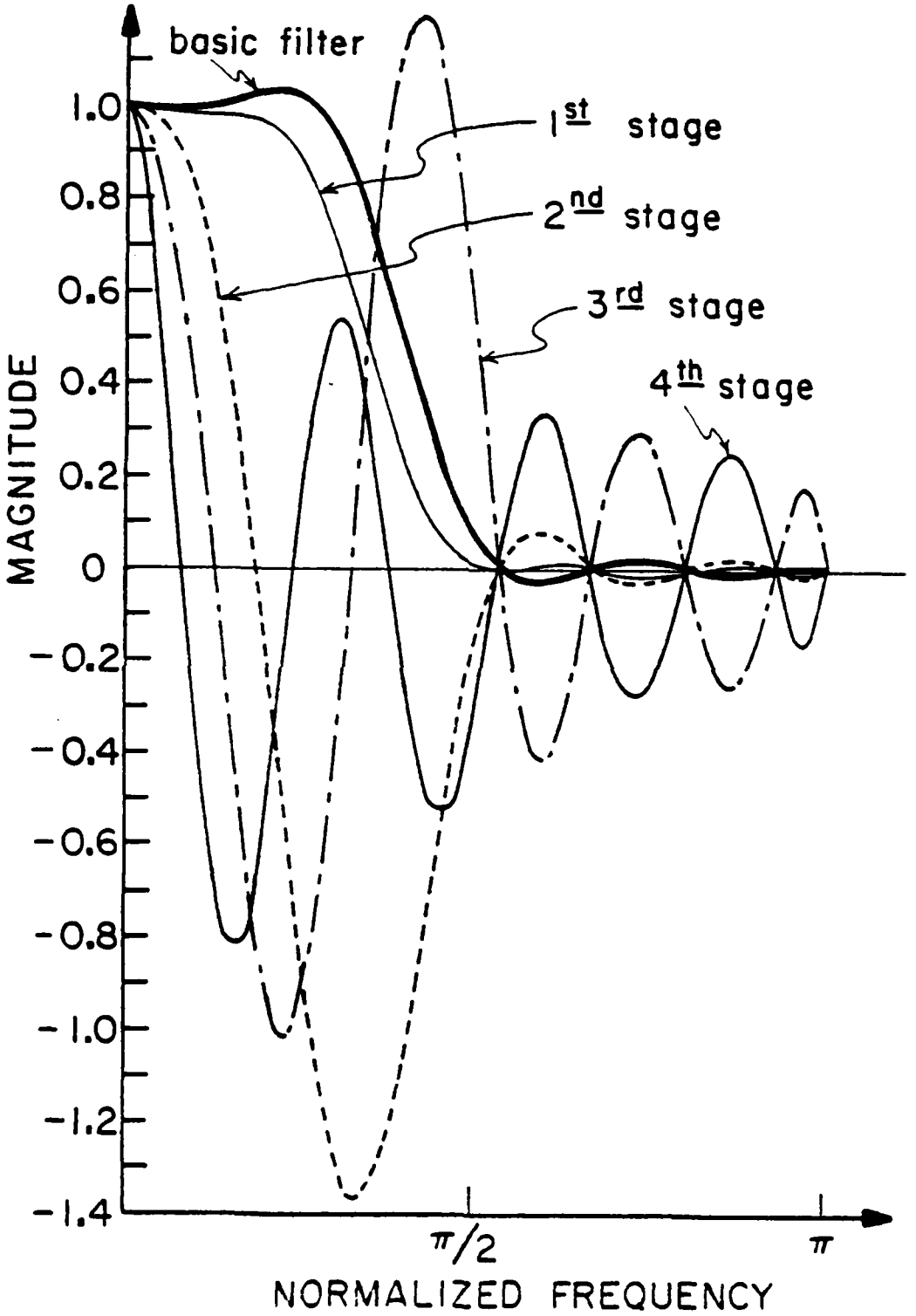


Figure 6-7. SVD-expanded filter frequency response

filter is obtained by substituting $(\frac{z+z^{-1}}{2})$ in Eq. (6-34) by the transformation

$$\frac{z_1+z_2^{-1}}{2} = \sum_{k=0}^{P} A_k^c \left(\frac{z_1+z_1^{-1}}{2}\right)^k$$
$$\frac{z_2+z_2^{-1}}{2} = \sum_{k=0}^{P} A_k^r \left(\frac{z_2+z_2^{-1}}{2}\right)^k \qquad (6-35)$$

If the basic filter is linear phase with cutoff frequencies, u_{c_1} and u_{c_2} along the horizontal and vertical axes, respectively, and if one is interested in changing the cutoff frequencies to β_{c_1} and β_{c_2}, basically the transformation is performed as it was in the one-dimensional case.

An example of the first-order transformation on the SVD/SGK convolution filter is shown in Fig. 6-8. The basic filter has quadrilateral symmetry with cutoff frequencies $u_{c_1} = u_{c_2} = 0.7097$. A perspective view of the frequency response is shown in Fig. 6-9. Unless stated otherwise, $u_{c_1} = u_{c_2} = u_c$ and $\beta_{c_1} = \beta_{c_2} = \beta_c$ are assumed for the transformation. Figure 6-8 shows the cross-sectional view of the frequency response on the horizontal axis. In the case of $u_c \leq \beta_c$, the transformation works quite adequately, but not for $u_c > \beta_c$. Table 6-4 summarizes the filter parameters and the measured and desired cutoff frequencies. Figure 6-10 shows another basic filter, which also possesses quadrilateral symmetry with cutoff frequency $u_c = 1.1266$.

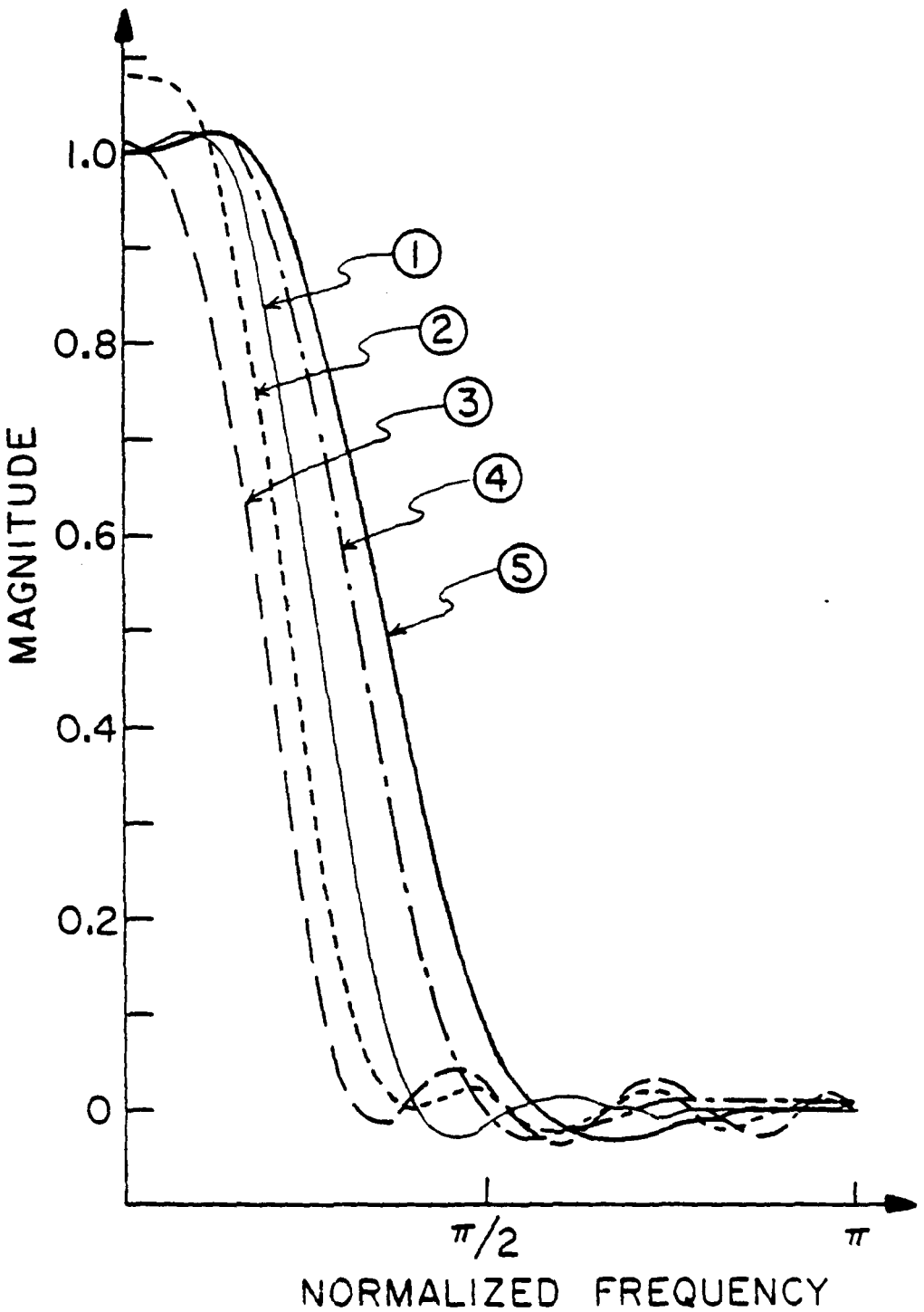


Figure 6-8. First-order transformation on the SVD/SGK convolution filter (horizontal)

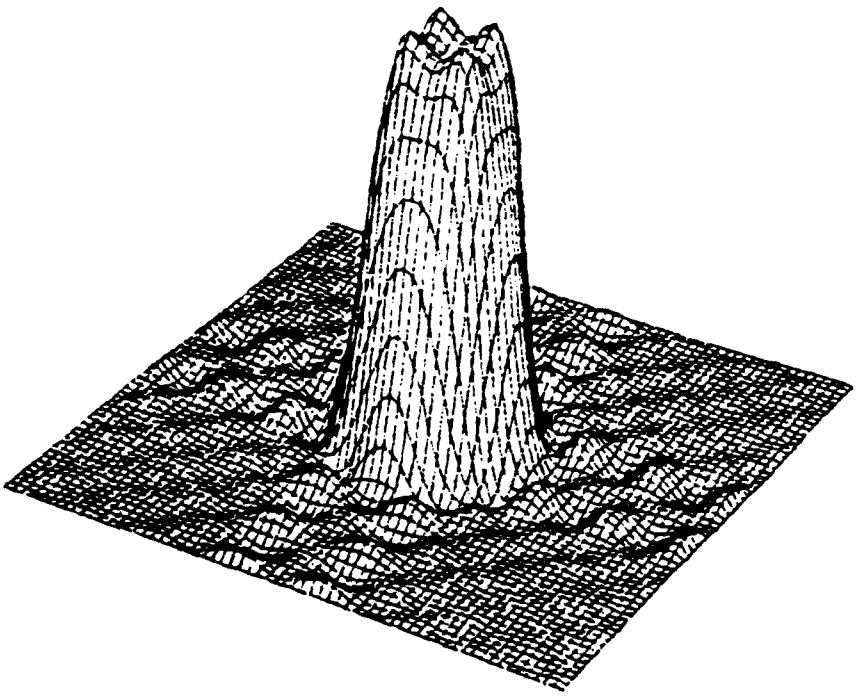


Figure 6-9. Perspective view of the frequency response of the basic filter

TABLE 6-4

List of the transformed SVD/SGK convolution filters and their cutoff frequencies using first-order transformation

Filter	Δ_1	Δ_2	Δf	Cutoff Frequency (Horizontal)		R(%)
				Desired	Measured	
1*	0.25	0.02	0.272	0.7097	0.7097	
2	0.08	0.04	0.188	0.6500	0.6056	-8.4
3	0.01	0.08	0.272	0.5800	0.4684	-18.3
4	0.02	0.04	0.241	0.8500	0.8515	19.8
5	0.02	0.03	0.283	0.9500	0.9520	33.9

*Basic Filter

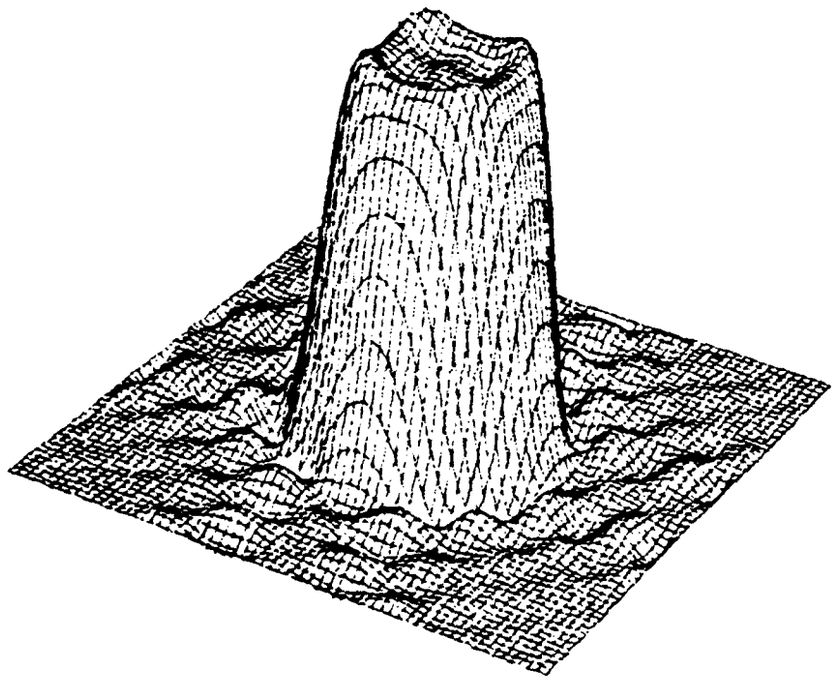


Figure 6-10. Perspective view of the frequency response of the basic filter

The results of the first- and second-order transformations are shown in Figures 6-11 and 6-12. Figure 6-11 corresponds to the horizontal, and Fig. 6-12 corresponds to the diagonal direction. Comparison of the first-order and second-order transformation shows that the second-order transformation yields far superior results (See Table 6-5).

Another interesting transformation is the case of $\beta_{c_1} \neq \beta_{c_2}$. An example of changing the cutoff frequency so that $u_{c_1} < \beta_{c_1}$ and $u_{c_2} > \beta_{c_2}$, is presented in Fig. 6-13, and the resulting filter perspective view of the frequency response is given in Fig. 6-14. In this experiment, the second-order transformation is used with the same basic filter shown in Fig. 6-10.

6.3 Fixed-Point Implementation

Once again, it is of great interest to implement a variable cutoff SVD/SGK convolution filter with special-purpose fixed-point arithmetic hardware. Since the transformation is mainly concerned with the cutoff frequency of the transformed filter, the effect of fixed-point implementation on the cutoff frequency is significant. Experimental evidence shows that a lowpass filter with filter coefficients rounded to 16 bits is sufficient for both first- and second-order transformation. The frequency responses with different word-length for filter coefficient quantization and with no rounding are

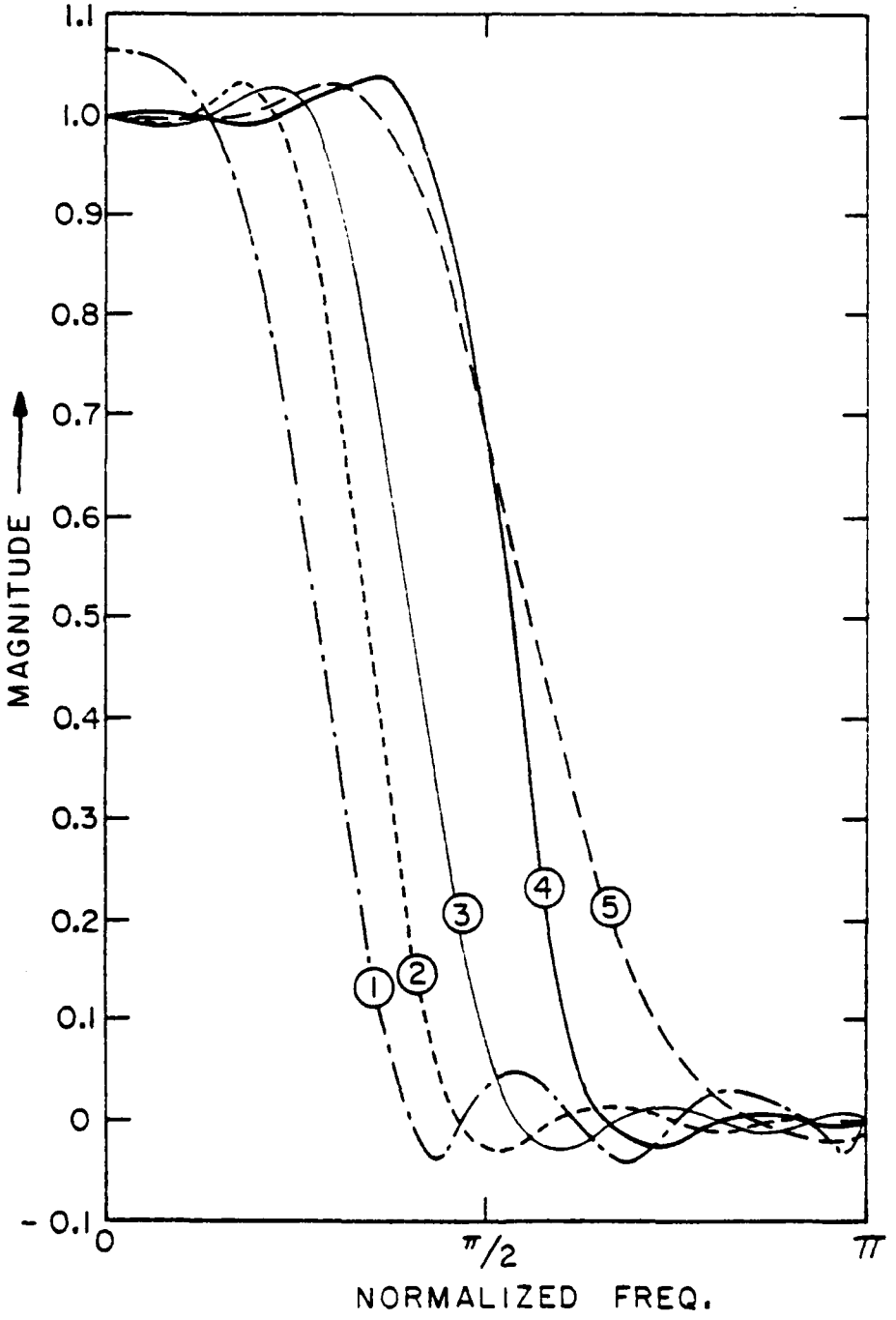


Figure 6-11. Transformation on the SVD/SGK convolution filter (horizontal)

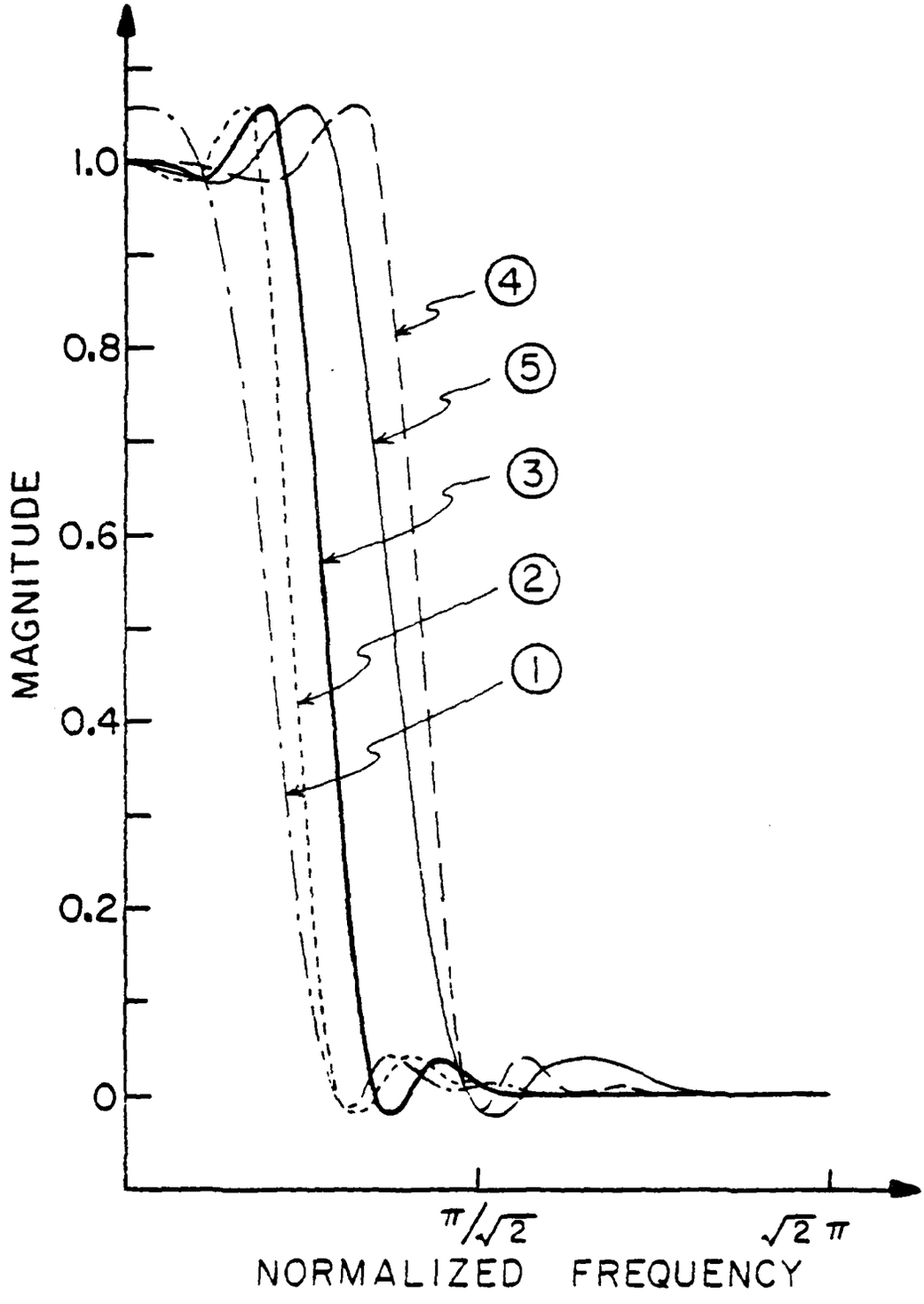


Figure 6-12. Transformation on the SVD/SGK convolution filter (diagonal)

TABLE 6-5

List of the transformed SVD/SGK convolution filters and their cutoff frequencies

Filter	Horizontal			Diagonal			Cutoff Frequency		R(%)	Type
	Δ_1	Δ_2	Δf	Δ_1	Δ_2	Δf	Desired	Measured		
1	0.06	0.09	0.470	0.06	0.06	0.281	0.95	0.7466	-15.7	First-order
2	0.04	0.02	0.262	0.08	0.08	0.187	0.95	0.9511	-15.7	Second-order
3	0.04	0.04	0.278	0.08	0.06	0.207	1.1266	1.1266		Basic Filter
4	0.03	0.04	0.269	0.07	0.04	0.207	1.5446	1.5439	37.1	Second-order
5	0.04	0.04	0.283	0.08	0.04	0.296	1.5446	1.5467	37.1	First-order

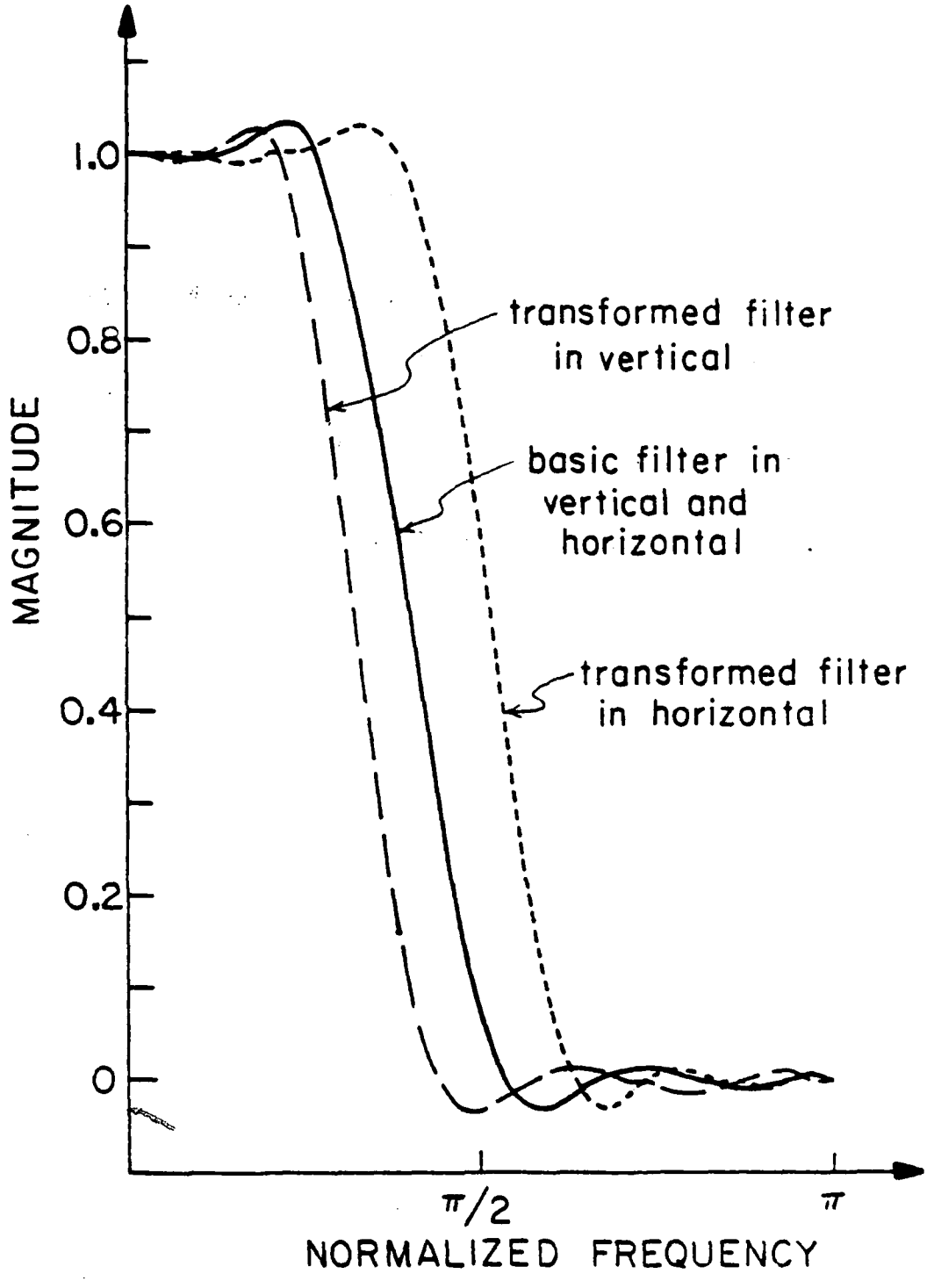


Figure 6-13. Second-order transformation for $\beta_{c_1} \neq \beta_{c_2}$ (horizontal)

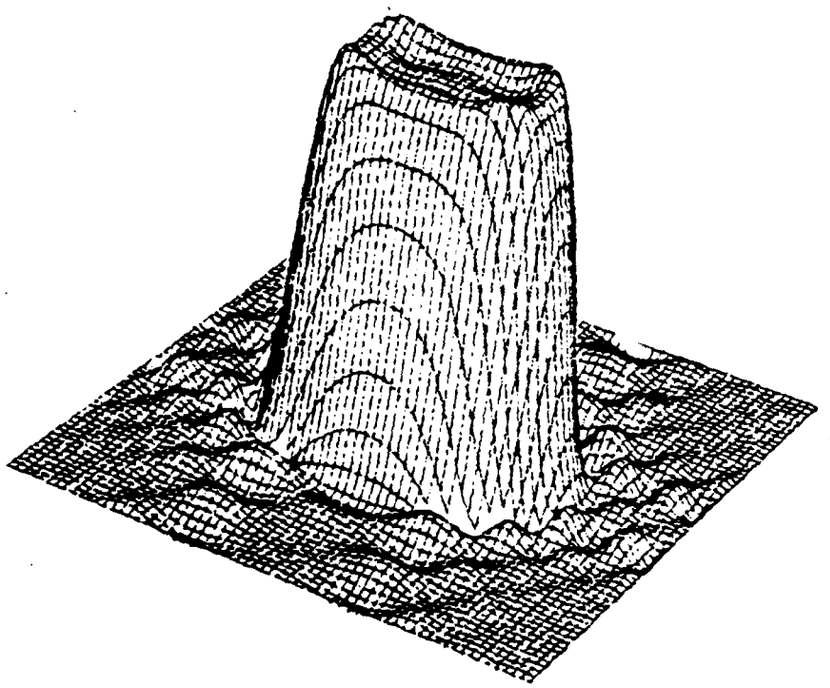


Figure 6-14. Perspective view of the frequency response of the transformed filter with $\beta_{c_1} \neq \beta_{c_2}$

plotted in Figures 6-15 and 6-16. In the case of first-order transformation, the response for the 8-bit case deviates from the ideal response significantly at the beginning of the passband, whereas no visible errors are seen anywhere in the stopband. In the case of the second-order transformation, the response for the 12-bit case shows the same characteristics. But the responses for 16-bit for both first- and second-order transformations are almost the same as the ideal. This experiment shows that, for a lowpass filter, the cascade form is highly sensitive to inaccurate coefficients in the passband, but the behavior in the stopband is much less sensitive. In addition, the second-order transformation requires a greater word-length to quantize the filter coefficients than the first-order transformation does. The basic filter was the same as shown in Fig. 6-10.

In order to investigate the roundoff noise effect on the fixed-point implementation of the variable cutoff SVD/SGK convolution filter, the random number array with a size of 46x46 was used again as an input. The correlation coefficient was 0.95. Basically, the same scaling procedure was used to prevent overflow, and the suboptimal ordering algorithm of Chapter 4 to minimize the roundoff noise. Theoretical estimates of the roundoff noise (standard deviation), based on the noise formula derived in Chapter 3, were computed and compared with the measured

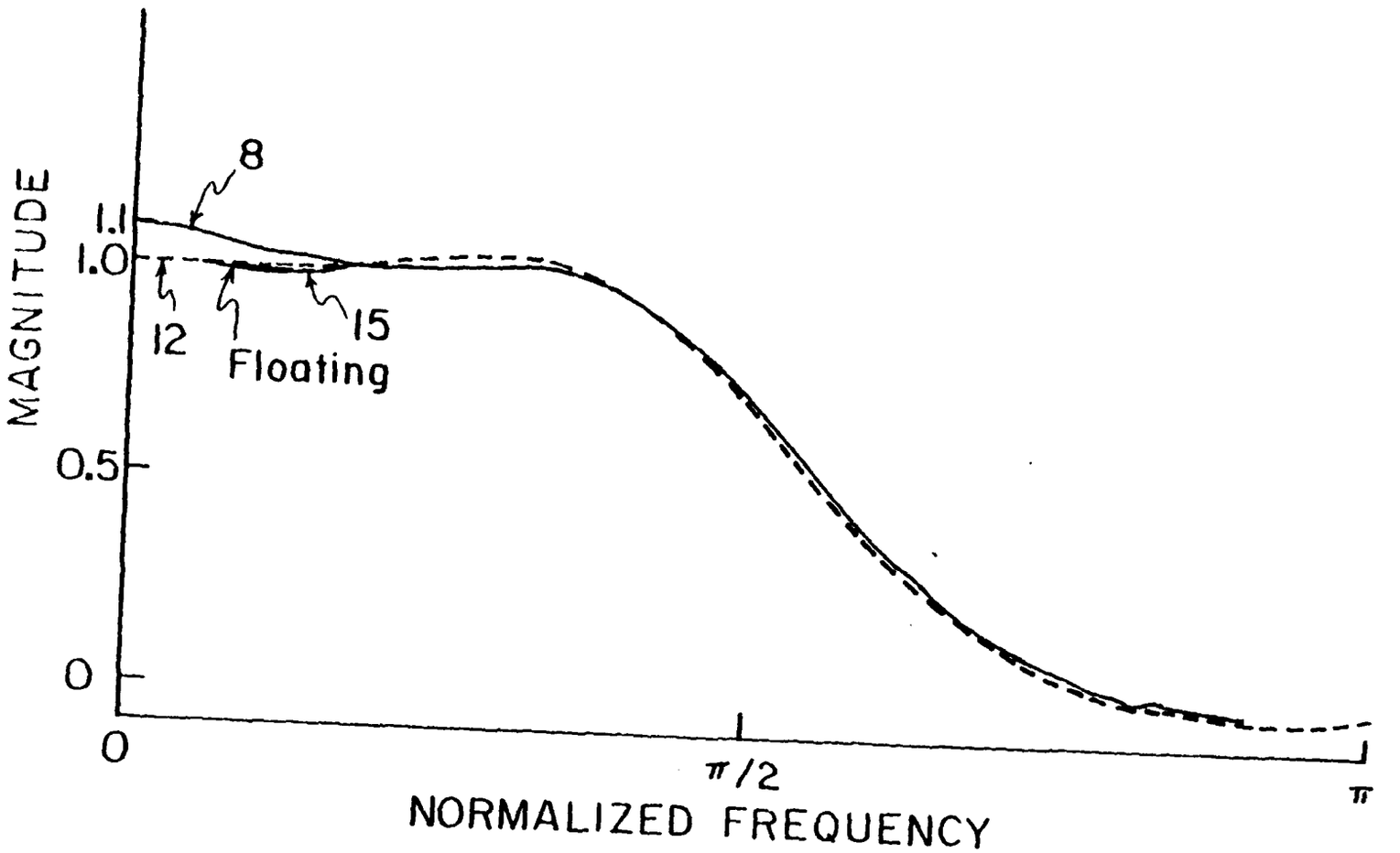


Figure 6-15. The effect of filter coefficient quantization with first-order transformation (horizontal)

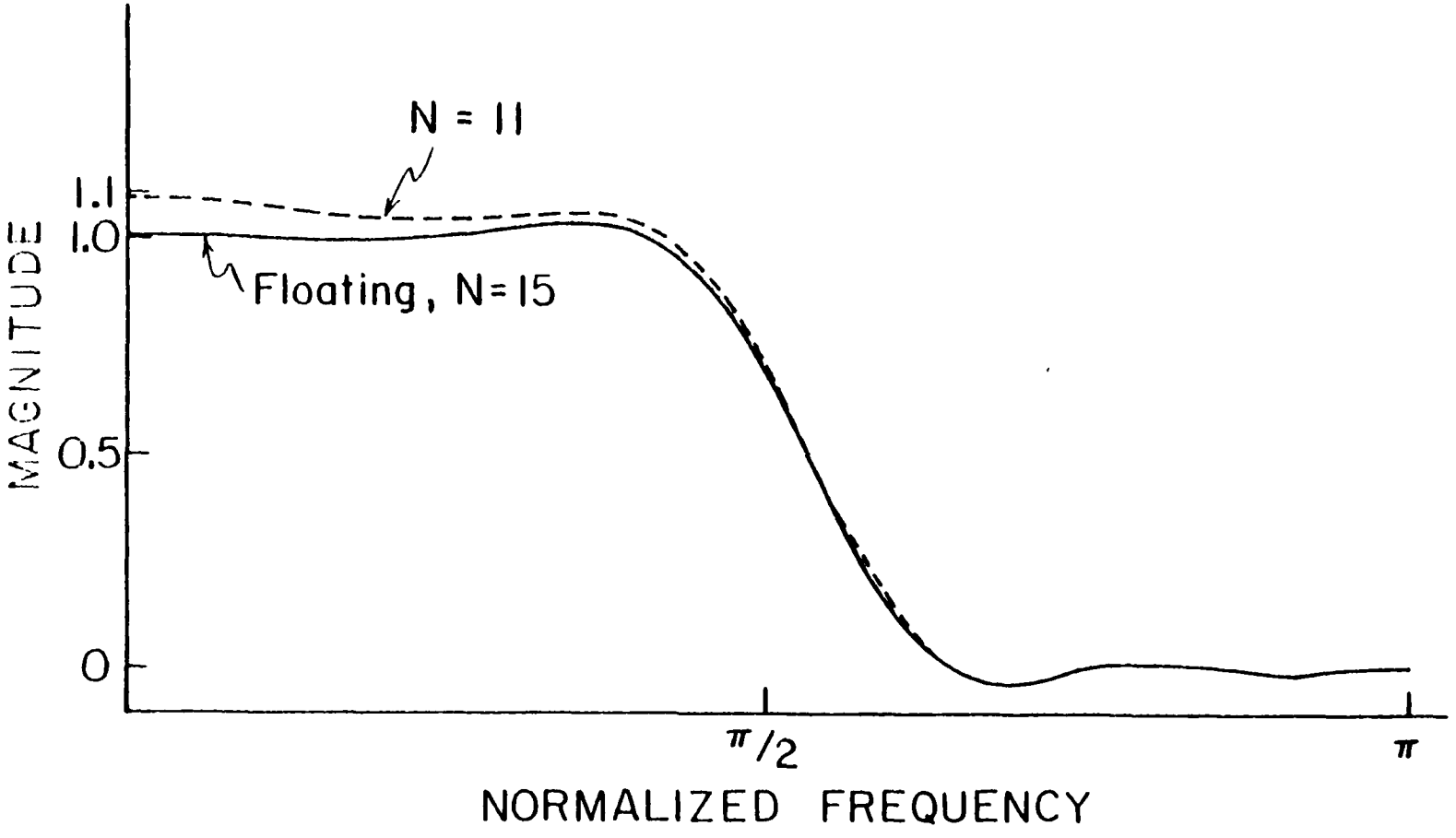


Figure 6-16. The effect of filter coefficient quantization with second-order transformation (horizontal)

values. Table 6-6 summarizes the results. In this experiment, we assumed that M = 16 bits and that only one rounding operation is performed within the SGK filter. Excellent agreement between the two values is observed. It is believed, again, that M = 16 and N = 12 are sufficient to achieve less than 1 % NMSE for both first- and second-order transformations. Surprisingly, it is noted that the required word-length for the variable SVD/SGK convolution filter is the same as required that in the SVD/SGK convolution filter.

6.4 Conclusion

In this chapter, attempts have been made to develop a design technique for variable cutoff SVD/SGK convolution filters. We considered first- and second-order transformations. Second-order transformation, in general, exhibits better results, but it inherently limits the range of transformation. It has been shown that second-order transformation is still possible if the specified constraint is relaxed. But the price paid for relaxing the constraint is degradation of the frequency characteristic of the transformed filter. In addition, the second-order transformation doubles the size of the transformed filter. The problem of implementing the transformed filter in fixed-point arithmetic was also discussed. For a lowpass filter, it was shown that the cascade form is highly

TABLE 6-6

Standard deviation of the transformed filter caused by rounding operation

M=16

N	Theory	Experiment	NMSE(%)
8	0.337×10^{-2}	0.548×10^{-2}	2.10
10	0.841×10^{-3}	0.119×10^{-2}	1.21
12	0.211×10^{-3}	0.295×10^{-3}	0.47
14	0.526×10^{-4}	0.768×10^{-4}	0.32
16	0.132×10^{-4}	0.203×10^{-4}	0.08

First-Order Transformation

M=16

N	Theory	Experiment	NMSE(%)
8	0.379x10	0.925x10	16.23
10	0.946x10	0.107x10	1.14
12	0.237x10	0.314x10	0.53
14	0.592x10	0.743x10	0.2
16	0.148x10	0.305x10	0.13

Second-Order Transformation

sensitive to inaccurate coefficients in the passband, but not in the stop band. Finally, it is believed that M = 16 and N = 12 are sufficient to obtain less than 1 % NMSE in most practical cases.

CHAPTER 7

SUMMARY AND FUTURE WORK

In this dissertaion, attempts have been made to describe a novel architecture for performing two-dimensional convolution with a minimum amount of hardware using the concept of sequential SGK convolution.

The singular value decomposition of an impulse response of a two-dimensional FIR filter has proven to be useful in designing two-dimensional approximating FIR filters that can be implemented as a cascade of 3x1 convolution operators. The usefulness of the SVD has been demonstrated by noting a trade off between approximation error and computational speed. The SVD/SGK convolution approach is particularly attractive when one is interested in implementing a two-dimensional convolution with a digital image display system. An approach to implementation for SVD/SGK convolution that employs a small set of relatively simple digital circuits has been described. It has been demonstrated that the statistical approach is very useful to analyze effects of finite word-length in representing filter coefficients and signal

magnitudes. A theoretical formula for predicting the total roundoff noise has been derived and confirmed experimentally.

Two important issues involving the implementation of a digital filter as a cascade of second-order filters, scaling and section ordering for SVD/SGK convolution, were also considered. We have shown how the algorithm available in the domain of one-dimensional signal processing can be extended to two-dimensional signal processing. One interesting result is that roundoff noise can be reduced by interlacing row and column oriented elementary second-order filters.

Experimental results dealing with image convolution show that 12 bits are required for memory storage (the most expensive part in image display systems), and 16 bits are needed for filter coefficient quantization if one desires to get results indistinguishable from the output using full precision. These features can be reduced if one allows some distortion in the image outputs. It has been shown that the quality of an SVD/SGK processed output is improved by resetting the output mean to be equal to the input mean. Since it is impractical to compute the output mean, a simple algorithm for resetting the output mean was proposed. The effectiveness of the proposed algorithm has been demonstrated visually.

It has been shown that parametric modification of the cutoff frequency of a filter is possible with transformation. Basically, the approach developed in the one-dimensional case by Oppenheim et al. [7-1] was used. A detailed analysis for first- and second-order transformations was made, and several design examples were presented. In the first-order transformation, the transformation could not properly preserve the frequency response of basic filter as the degree of transformation increased. In the second-order transformation, due to inherent characteristics of trigonometric functions in the transformation, the transformation works only in a limited range. In other words, it is impossible to change the cutoff frequency of the basic filter arbitrarily. But the resulting transformed filter shows a frequency response almost identical to the basic filter within the transformation range. It was found that, by relaxing the specified constraint, arbitrary variation of the cutoff frequency of the basic filter with the second-order transformation is still feasible. But a degradation of the transformed filter frequency response was also observed. It is believed that with both transformations, there should be a trade off between the degree of transformation and the preservation of the frequency response of the basic filter. Finally, it was found experimently that 12 bits for the accumulator memory and 16 bits for the filter coefficients

are also sufficient to limit quantization and roundoff noise effects to less than 1 % NMSE in both the first- and second-order transformations.

Several problems are worthy of further investigation. If one is particularly interested in implementation speed, the SGK convolution approach is always faster than the SVD/SGK convolution approach. Comparison of the processing frame cycle required for SGK and SVD/SGK convolution shows that only Q frame cycles are needed with SGK convolution, while the SVD/SGK convolution requires 2KQ frame cycles when the size of impulse response is (2Q+1)x(2Q+1) and K is the number of singular values employed. Finding a simple analytical design procedure for an SGK convolution filter is still problem. An alternative to one-dimensional SVD/SGK convolution is to use two-dimensional SVD/SGK convolution. Two-dimensional SVD/SGK convolution reduces the computational speed by a factor of 2. In this case, the scaling procedure should be carefully chosen to use the full dynamic range of the given word-length. It is also expected that two-dimensional SVD/SGK convolution requires a greater word-length.

In the previous approach to the parametric design, we used one-dimensional methods to design a variable cutoff SVD/SGK convolution filter. In addition, the basic filter was restricted to be linear phase. A generalized design

technique for the two-dimensional variable cutoff digital filter would be useful. Finally, entending SGK or SVD/SGK convolution to the recursive approach would also help solve the two-dimensional signal processing problem.

APPENDIX A

Relation between ε_k and ε_l Error

Let $\underline{G}$ and $\underline{\hat{G}}_{SVD}$ be the output array of size MxM as defined in Section 5-2. Their variances σ_g^2 and $\sigma_{\hat{g}}^2$ are given by

$$\sigma_g^2 = \frac{\sum_i \sum_j |G(i,j) - m_g|^2}{M^2} \qquad (A-1)$$

$$\sigma_{\hat{g}}^2 = \frac{\sum_i \sum_j |\hat{G}_{SVD}(i,j) - m_{\hat{g}}|^2}{M^2} \qquad (A-2)$$

where m_g and $m_{\hat{g}}$ denote the mean of the output array $\underline{G}$ and $\underline{\hat{G}}_{SVD}$, respectively. If the error array $\underline{E}$ is defined as

$$\underline{E} = \underline{G} - \underline{\hat{G}}_{SVD} \qquad (A-3)$$

then, its variance is given by

$$\sigma_e^2 = \frac{\sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j) - m_g + m_{\hat{g}}|^2}{M^2} \qquad (A-4)$$

After some algebraic manipulation, it can be shown that

$$M^2\sigma_E^2 = \sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2 - M^2(m_g - m_{\hat{g}})^2 \qquad \text{(A-5)}$$

and

$$M^2\sigma_g^2 = \sum_i \sum_j |G(i,j)|^2 - M^2 m_g \qquad \text{(A-6)}$$

Dividing Eq. (A-5) by Eq. (A-6), we obtain the following relation.

$$\frac{\sigma_e^2}{\sigma_g^2} = \frac{\sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2}{\sum_i \sum_j |G(i,j)|^2} \cdot \left[\frac{1 - \dfrac{\varepsilon_m^2 M^2}{\sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2}}{1 - \dfrac{m_g^2 M^2}{\sum_i \sum_j |G(i,j)|^2}} \right] \qquad \text{(A-7)}$$

where $\varepsilon_m = m_g - m_{\hat{g}}$. Note that

$$\varepsilon_1^2 = \frac{\sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2}{\sum_i \sum_j |G(i,j)|^2} \qquad \text{(A-8)}$$

and if we let

$$\alpha = \frac{1 - \dfrac{\varepsilon_m^2 M^2}{\sum_i \sum_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2}}{1 - \dfrac{m_g^2 M^2}{\sum_i \sum_j |G(i,j)|^2}} \qquad \text{(A-9)}$$

then,

$$\frac{\sigma_e^2}{\sigma_g^2} = \varepsilon_1^2 \cdot \alpha \tag{A-10}$$

Suppose $F(k,\ell)$ is an input array and m_f denotes its mean, then σ_e^2 and σ_g^2 can be also expressed by

$$\sigma_e^2 = \sum_{i'}\sum_{j'}\sum_{i}\sum_{j} [\hat{H}(i,j)-H_{SVD}(i,j)]\Lambda(i-i',j-j') \cdot [H(i',j')-\hat{H}_{SVD}(i',j')] \tag{A-11}$$

and

$$\sigma_g^2 = \sum_{i'}\sum_{j'}\sum_{i}\sum_{j} H(i,j)\Lambda(i,i',j-j')H(i',j') \tag{A-12}$$

where

$$\Lambda(i,j) = \sum_{k}\sum_{\ell}[F(k,\ell)-m_f][F(k+i,\ell+j)-m_f] \tag{A-13}$$

Combining Eqs. (A-11) and (A-12), and substituting into Eq. (A-10), yields

$$\varepsilon_1^2 = \left[\frac{\sum_{i'}\sum_{j'}\sum_{i}\sum_{j} [H(i,j)-\hat{H}_{SVD}(i,j)]\Lambda(i-i',j,j')}{\sum_{i'}\sum_{j'}\sum_{i}\sum_{j} H(i,j)\Lambda(i-i',j-j')} \cdot \frac{[H(i',j')-\hat{H}_{SVD}(i',j')]}{H(i',j')} \right] \cdot \alpha \tag{A-14}$$

But the first term in Eq. (A-11) is equivalent to ε_k^2. Thus,

$$\varepsilon_l^2 = \varepsilon_k^2 \cdot \alpha \tag{A-15}$$

or

$$\varepsilon_l = E_k \cdot \left[\frac{1 - \dfrac{\varepsilon_m^2 M^2}{\sum\limits_i \sum\limits_j |G(i,j) - \hat{G}_{SVD}(i,j)|^2}}{1 - \dfrac{m_g^2 M^2}{\sum\limits_i \sum\limits_j |G(i,j)|^2}} \right]^{\frac{1}{2}} \tag{A-16}$$

It is clear that ε_l will decrease if one can set ε_m close to zero.

APPENDIX B

Relation between Basic Filter Coefficients and Transformed Filter Coefficients

Let h(n) for n = 0,1,...,2Q+1 represents an impulse response of the basic filter and a(n) for n = 0,1,...,2QP+1 represents the impulse response of the transformed filter. It was shown that the Fourier transform of a symmetrical filter can be expressed as

$$h(e^{ju}) = e^{-juQ} \sum_{n=0}^{Q} b(n)(\cos u)^n \qquad \text{(B-1)}$$

The new coefficient b(n) is obtained from the Chebyshev polynomial recursion formula. The relation between h(n) and b(n) is given by

$$\begin{bmatrix} b(0) \\ b(1) \\ b(2) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \end{bmatrix} \qquad \text{(B-2a)}$$

for Q = 1 and

$$\begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ b(3) \\ b(4) \end{bmatrix} = \begin{bmatrix} -1 & 0 & 1 & 0 & -1 \\ 0 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 & 0 \\ -1 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} h(0) \\ h(1) \\ h(2) \\ h(3) \\ h(4) \end{bmatrix} \qquad \text{(B-2b)}$$

for $Q = 2$. By expanding Eq. (6-9) with $P = 1,2$, the coefficient $a(n)$ can be expressed in terms of $b(n)$ and the control parameter A_K.

If $P = 1$, then

$$\begin{bmatrix} a(0) \\ a(1) \\ a(2) \end{bmatrix} = \begin{bmatrix} 0 & \frac{A_1}{2} & 0 \\ \frac{1}{2} & A_0 & \frac{1}{2} \\ 0 & \frac{A_1}{2} & 0 \end{bmatrix} \begin{bmatrix} b(0) \\ b(1) \\ b(2) \end{bmatrix} \qquad \text{(B-3a)}$$

for $Q = 1$ and

$$\begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \\ a(4) \end{bmatrix} = \begin{bmatrix} 0 & 0 & \frac{A_1}{4} & 0 & 0 \\ 0 & \frac{A_1}{4} & A_0A_1 & \frac{A_1}{4} & 0 \\ \frac{1}{2} & \frac{A_0}{2} & A_0^2 + \frac{A_1^2}{2} & \frac{A_0}{2} & \frac{1}{2} \\ 0 & \frac{A_1}{4} & A_0A_1 & \frac{A_1}{4} & 0 \\ 0 & 0 & \frac{A_1}{4} & 0 & 0 \end{bmatrix} \begin{bmatrix} b(0) \\ b(1) \\ b(2) \\ b(3) \\ b(4) \end{bmatrix} \qquad \text{(B-3b)}$$

for $Q = 2$, where $A_0 + A_1 = 1$ for $u_c \leq \beta_c$ and $-A_0 + A_1 = 1$ for $u_c > \beta_c$.

If P = 2, the relation is given by

$$\begin{bmatrix} a(1) \\ a(1) \\ a(2) \\ a(3) \\ a(4) \end{bmatrix} = \begin{bmatrix} 0 & \frac{A_2}{4} & 0 \\ 0 & \frac{A_1}{2} & 0 \\ \frac{1}{2} & A_0 + \frac{A_2}{2} & \frac{1}{2} \\ 0 & \frac{A_1}{2} & 0 \\ 0 & \frac{A_2}{4} & 0 \end{bmatrix} \begin{bmatrix} b(0) \\ \\ b(1) \\ \\ b(2) \end{bmatrix} \qquad \text{(P-4a)}$$

for Q = 1 and

$$\begin{bmatrix} a(0) \\ a(1) \\ a(2) \\ a(3) \\ a(4) \\ a(5) \\ a(6) \\ a(7) \\ a(8) \end{bmatrix} = \begin{bmatrix} 0 & 0 & m_1 & 0 & 0 \\ 0 & 0 & m_2 & 0 & 0 \\ 0 & m_3 & m_4 & m_3 & 0 \\ 0 & m_5 & m_6 & m_5 & 0 \\ \frac{1}{2} & m_7 & m_8 & m_7 & \frac{1}{2} \\ 0 & m_5 & m_6 & m_5 & 0 \\ 0 & m_3 & m_4 & m_3 & 0 \\ 0 & 0 & m_2 & 0 & 0 \\ 0 & 0 & m_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} b(0) \\ \\ b(1) \\ \\ b(2) \\ \\ b(3) \\ \\ b(4) \end{bmatrix} \qquad \text{(P-4b)}$$

where

$$\begin{aligned} m_1 &= \frac{A_2^2}{16} \\ m_2 &= \frac{A_1 A_2}{4} \\ m_3 &= \frac{A_2}{8} \\ m_4 &= \frac{A_1^2}{4} + \frac{A_0 A_2}{2} + \frac{A_2^2}{4} \\ m_5 &= \frac{A_1}{4} \end{aligned} \qquad \text{(P-4c)}$$

$$m_6 = A_0A_1 + \frac{3}{4} A_1A_2$$

$$m_7 = \frac{A_0}{2} + \frac{A_2}{4}$$

$$m_8 = A_0^2 + \frac{A_1}{2} + A_0A_2 + \frac{3}{8} A_2^2$$

for Q = 2. But, $A_0 = -A_2$ and $A_1 = 1$ when P = 2.

Therefore, the relation between a(n) and h(n) can be obtained directly by substituting Eq. (B-2) into Eq. (B-3) for first-order transformation and into Eq. (B-4) for second-order transformation.

APPENDIX C

Derivation of Eqs. (6-26) and (6-27)

The second-order transformation can be characterized by

$$\cos u = A_0 + A_1\cos\beta + A_2\cos^2\beta \tag{C-1}$$

By imposing the constraints at 0 and π, we obtain

$$\begin{aligned} 1 &= A_0 + A_1 + A_2 \\ -1 &= A_0 - A_1 + A_2 \end{aligned} \tag{C-2}$$

Equation (C-2) immediately leads to the relations

$$A_0 = -A_2 \tag{C-3a}$$

$$A_1 = 1 \tag{C-3b}$$

Substitution of Eq. (C-3) into Eq. (C-1) results in

$$\cos u = A_0 + \cos\beta - A_0\cos^2\beta \tag{C-4}$$

The range of A_0, satisfying,

$$-1 \leq f(x) \leq 1 \tag{C-5}$$

will ensure that $|\cos u| \leq 1$ where $f(x) = A_0 + x - A_0 x^2$, $x = \cos\beta$ but $-1 \leq x \leq 1$. Note that $f(x)$ is a quadratic

function in x and always passes through two points, (-1, -1) and (1,1). The only case for which the condition of Eq. (C-5) is being satisfied is shown in Fig. C-1. That is equivalent to solving

$$\frac{1}{2A_0} \geq 1 \quad , \quad A_0 > 0 \tag{C-6a}$$

$$\frac{1}{2A_0} < -1 \quad , \quad A_0 < 0 \tag{C-6b}$$

Equation (C-6) gives the range of A_0 such that

$$-\frac{1}{2} \leq A_0 \leq \frac{1}{2} \tag{C-7}$$

but, $0 \leq A_0 \leq \frac{1}{2}$ corresponds to the case of $u_c \leq \beta_c$ and $-\frac{1}{2} \leq A_0 \leq 0$ corresponds to the case of $u_c > \beta_c$, respectively. $A_0 = 0$ means that the transformed filter is identical to the basic filter.

The relation between u_c and β_c is obtained by solving Eq. (C-7), which is a quadratic function in cos β. That is

$$\cos\beta_c = \frac{1 \pm \sqrt{1-4A_0(\cos u_c - A_0)}}{2A_0} \tag{C-8}$$

But the plus sign in Eq. (C-8) is discarded since $|\cos\beta| \leq 1$.

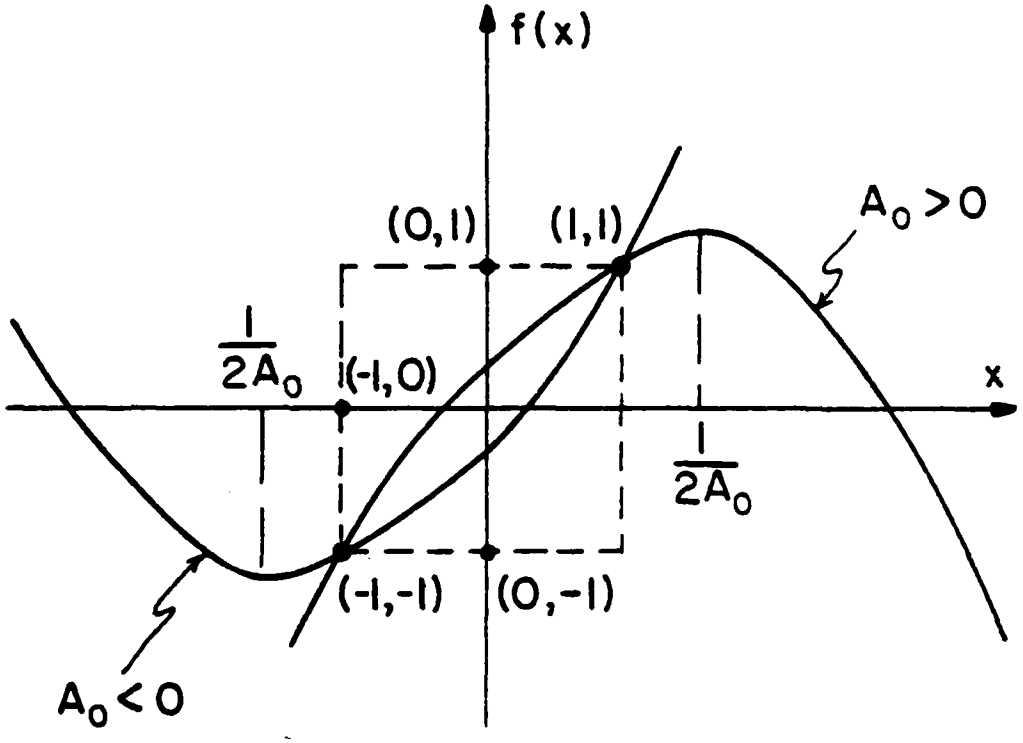


Figure C-1. Illustration of condition

REFERENCES

[1-1] L.B. Jackson, J.F. Kaiser, and H.S. McDonald, "An Approach to the Implementation of Digital Filers," IEEE Transactions on Audio and Flectroacoustics, Vol. AU-16, No. 3, September 1968, pp. 413-421.

[1-2] W.K. Pratt, "An Intelligent Image Processing Display Terminal," Proc. SPIE Tech. Symp., Vol. 27, San Diego, Calif., August 1979.

[1-3] H.C. Andrews, "Digital Image Processing," IEEE Spectrum, Vol. 16, No. 4, April 1979, pp. 38-49.

[1-4] J.F. Abramatic and O.D. Faugeras, "Design of Two-Dimensional FIR Filters from Small Generating Kernels," Proc. IEEE Conference on Pattern Recognition and Image Processing, Chicago, Illinois, May 1978.

[1-5] W.F.W. Mecklenbrauker and R.M. Mersereau, "McClellan Transformations for Two-Dimensional Digital Filtering: II-Implementation," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 7, July 1976, pp. 414-422.

[1-6] A.V. Oppenheim, W.F.G. Mecklenbrauker, and R.M. Mersereau, "Variable Cutoff Linear Phase Digital Filters," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 4, April 1976, pp. 199-203.

[2-1] T.S. Huang, Picture Processing and Digital Filtering Topics in Applied Physics, Springer-Verlag, New York, 1975.

[2-2] M.P. Ekstrom and S.K. Mitra, Two-Dimensional Digital Signal Processing, Dowden-Hutchinson and Ross, Inc., 1978.

[2-3] J.W. Cooley and J.W. Tukey, "An Algorithm for the Machine Calculation of Fourier Series," Mathematics of Computation, Vol. 19, April 1965, pp. 297-301.

[2-4] W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

[2-5] R.M. Mersereau and D.E. Dudgeon, "Two-Dimensional Digital Filtering," Proc. IEEE, Vol. 63, No.4, April 1975, pp. 610-623.

[2-6] B.R. Hunt, "Minimizing the Computation Time by Using the Techniques of Sectioning for Digital Filtering of Pictures," IEEE Transactions on Computer, Vol. C-21, No. 11, Nov. 1972, pp. 1219-1222.

[2-7] T.G. Stockham, Jr., "High Speed Convolution and Correlation," Proc. Spring Joint Computer Conference, 1966, pp. 229-233.

[2-8] J.W. Cooley, P. Lewis, and P.D. Welch, "The Finite Fourier Transform," IEEE Transactions on Audio and Electroacoustics, Vol. AF-17, No. 2, June 1969, pp. 77-86.

[2-9] J.H. McClellan, "The Design of Two-Dimensional Digital Filters by Transformations," 7th Annual Princeton Conference on Information Science and Systems, March 1973, pp. 247-251.

[2-10] R.M. Mersereau, W.F.G. Mechlenbrauker, and T.F. Quantieri, Jr., "McClellan Transformations for Two-Dimensional Digital Filtering: I-Design," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 7, July 1976, pp. 405-414.

[2-11] W.F.W. Mecklenbrauker and R.M. Mersereau, "McClellan Transformations for Two-Dimensional Digital Filtering: II-Implementation," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 7, July 1976, pp. 414-422.

[2-12] J.H. McClellan and D.S.K. Chan, "A 2-D FIR Filter Structure Derived from the Chebyshev Recursion," IEEE Transactions on Circuits and Systems, Vol. CAS-24, No. 7, July 1977, pp. 372-378.

[2-13] J.F. Abramatic and C.D. Faugeras, "Design of Two-Dimensional FIR Filters from "Small" Generating Kernels," Proc. IEEE Conf. on Pattern Recognition and Image Processing, Chicago, May 1978.

[2-14] J.F. Abramatic and O.D. Faugeras,"Suboptimal Chebyshev Design of 2-D FIR Filters from "Small" Generating Kernels," International Conf. on Digital Signal Processing, Florence, Sept. 1978.

[2-15] J.F. Abramatic, "Image Filtering by Sequential Convolution of "Small" Generating Kernels," Proc. of the IEEE International Symposium on Circuits and Systems, Tokyo, Japan, July 1979.

[2-16] G.H. Goulb and C. Reinsch, "Singular Value Decomposition and Least Square Solution," Numer. Math., Vol. 14, 1970, pp. 403-420.

[2-17] S. Treitel and J.L. Shanks, "The Design of Multistage Separable Planar Filters," IEEE Transactions on Geosci. Electron, Vol. GE-9, No. 1, Jan. 1971, pp. 10-27.

[2-18] R.E. Twogood and S.K. Mitra, "Computer-Aided Design of Separable Two-Dimensional Digital Filters," IEEE Transactions on Acoust., Speech, and Signal Processing, Vol. ASSP-25, No. 2, April 1977, pp. 165-169.

[2-19] B.R. Hunt,"A Matrix Theory Proof of the Discrete Convolution Theorm," IEEE Transactions on Audio and Electroacoustic, Vol. AU-19, No. 4, December 1971, pp. 285-288.

[2-20] H.C. Andrews and B.R. Hunt, Digital Image Restoration, Prentice-Hall, New Jersey, 1977.

[2-21] P. Lancaster, Theory of Matrices, Academic Press, New York, 1969.

[2-22] N.D. Mascarenhas and W.K. Pratt, "Digital Image Restoration Under a Regression Model," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 3, March 1975, pp. 252-266.

[2-23] W.K. Pratt, "An Intelligent Image Processing Display Terminal," Proc. SPIE Tech. Symp., Vol. 27, San Diego, Calif., August 1979.

[3-1] L.B. Jackson, "Roundoff Noise Analysis for Fixed-Point Digital Filter Realized in Cascade or Parallel Form," IEEE Transactions on Audio and Electroacoustics, Vol. AU-18, No. 2, June 1970, pp. 107-122.

[3-2] L.B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Ranges in Digital Filters," B.S.T.J., Vol. 49, No. 2, Feb. 1970, pp. 159-184.

[3-3] B. Liu, "Effect of Finite Word Length on the Accuracy of Digital Filters - A Review," IEEE Transactions on Circuit Theory, Vol. CT-18, No. 6, Nov. 1971, pp. 670-677.

[3-4] D.S.K. Chan and L.R. Rabiner, "Theory of Roundoff Noise in Cascade Realization of finite Impulse Response Digital Filters," B.S.T.J., Vol. 52, No. 3, March 1973, pp. 329-345.

[3-5] D.S.K. Chan and L.R. Rabiner, "Analysis of Quantization Errors in the Direct Form for Finite Form of Finite Impulse Response Digital Filters," IEEE Transactions on Audio and Electroacoustics, Vol. AU-21, No. 4, August 1973, pp. 354-366.

[3-6] A.V. Oppenheim and R.W. Schafer, Digital Signal Processing, Prentice-Hall, New Jersey, 1975.

[3-7] S.A. Trettler, Introduction to Discrete-Time Signal Processing, John Wiley and Sons, New York, 1976.

[3-8] R.E. Crochiere, "Digital Network Theory and its Application to the Analysis and Design of Digital Filter," Ph.D. Dissertation, Dept. of Elec. Eng., M.I.T., Cambridge, Mass., 1974.

[3-9] O. Hermann and H.W. Schuessler, "On the Accuracy Problem in the Design of New Recursion Digital Filters," Arch. Elek. Ubertragung, Vol. 24, Heft 11, Nov. 1970, pp. 525-526.

[4-1] W. Schussler, "On the Structure for Nonrecursive Digital Filters," Arch. Elek Ubertragung, Vol. 26, Heft 6, June 1972, pp. 255-258.

[4-2] D.S.K. Chan and L.R. Rabiner, "An Algorithm for Minimizing Roundoff Noise in Cascade Realization of Finite Impulse Response Digital Filters," B.S.T.J., Vol. 52, No. 3, March 1973, pp. 347-385.

[4-3] L.B. Jackson, "On the Interaction of Roundoff Noise and Dynamic Range in Digital Filters," B.S.T.J., Vol. 49, No. 2, Feb. 1970, pp. 159-184.

[4-4] L.R. Rabiner and B. Gold, Theory and Application of Digital Signal Processing, Prentice-Hall, New Jersey, 1975.

[5-1] W.K.Pratt, O.D. Faugeras and A. Gagalowicz, "Visual Discrimination of Stochastic Texture Field," IEEE Transactions on System, Man, and Cybernetics, Vol. SMC-8, No. 11, November 1978, pp. 796-804.

[5-2] J.B. Knowles and E.M. Olcayto, "Coefficient Accuracy and Digital Filter Response," IEEE Transactions on Circuit Theory, Vol. CT-15, No. 1, March 1968, pp. 31-41.

[5-3] J.F. Harris, Sr., "Image Evaluation and Restoration," J. Opt. Soc. Am., 56, No. 5, May 1966, pp. 569-574.

[5-4] W.K. Pratt, Digital Image Processing, Wiley-Interscience, New York, 1978.

[6-1] A.G. Constantinides, "Frequency Transformations for Digital Filters," Electron Letters, Vol. 3, No. 11, November 1967, pp. 487-489.

[6-2] A.G. Constantinides, "Frequency Transformations for Digital Filters," Electron Letters, Vol. 4, No. 7, April 1968, pp. 115-116.

[6-3] W. Schuessler and W. Winkelnkemper, "Variable Digital Filter," Arch. Electr. Ubertr, Vol. 24, No. 11, November 1970, pp. 524-525.

[6-4] A.V. Oppenheim, W.F.G. Mecklenbrauker, and R.M. Mersereau, "Variable Cutoff Linear Phase Digital Filters," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 4, April 1976, pp. 199-203.

[7-1] A.V. Oppenheim, W.F.G. Mecklenbrauker, and R.M. Mersereau, "Variable Cutoff Linear Phase Digital Filters," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 4, April 1976, pp. 199-203.